

SVEUČILIŠTE U ZAGREBU
FILOZOFSKI FAKULTET
ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE
ZNANOSTI

Ak. god. 2016/2017.

Dominik Pavičić

Kreiranje videoigara kao potpore u prezentiranju znanja

Diplomski rad

Mentorica: prof. dr. sc. Jadranka Lasić-Lazić

Zagreb, 2017.

Sažetak:

Cilj rada je predstaviti način na koji nastavnici mogu kreirati videoigre koje će im poslužiti kao alat za prezentaciju znanja te ujedno i predstaviti mogućnosti koje takav alat može ponuditi nastavnicima. Budući da takvo korištenje videoigara pripada području e-učenja, ponajprije se u radu predstavlja sam pojam e-učenja. Kroz rad će se pojasniti sama definicija pojma e-učenja, napraviti njegov povijesni pregled, te će se na temelju postojećih istraživanja predstaviti prednosti i nedostaci koji se javljaju prilikom korištenja e-učenja.

Nakon što je predstavljeno e-učenje, u radu će se pobliže razraditi Microsoft PowerPoint kao najkorišteniji alat za prezentaciju znanja. Budući da je cilj rada predstaviti alat koji će služiti kao alternativa PowerPointu, razradit će se paradigma korištenja PowerPointa kao alata za prezentiranje znanja, te principi na temelju kojih se kreiraju prezentacije. Potom će se prikazati prednosti i nedostaci PowerPointa kao alata za prezentiranje.

Na temelju razjašnjene definicije videoigara u obrazovanju, prikazat će se pregled dosadašnjeg korištenja videoigara u prezentiranju znanja i u samom obrazovanju kao posljednja točka koju treba razraditi prije nego se krene pojašnjavati kako se videoigre mogu koristiti kao alat za prezentaciju znanja. Ovdje će se pomnije prikazati kako su se do sada videoigre koristile u obrazovanju te koje prednosti i nedostatke njihovo korištenje donosi nastavnicima.

U istraživačkom dijelu rada će se ponajprije pojasniti podloga unutar igrinog pokretača Unity koja je neophodna za daljnje kreiranje videoigre bez znanja programiranja. Ovdje će se predstaviti sve skripte koje su napravljene za kreiranje prezentacijske videoigre kako bi se jasnije predstavile mogućnosti i ograničenja takvoga alata te kako bi se omogućilo njegovo daljnje proširenje.

Nakon toga će se razraditi upute kako se, uz prethodno navedenu podlogu, mogu kreirati videoigre za prezentiranje znanja te će, za svaku od mogućnosti alata, biti prikazan primjer. Pored toga, bit će predstavljene i prednosti i nedostaci korištenja ovoga alata za nastavnike.

Ključne riječi: Prezentiranje znanja, Prezentacijske videoigre, E-učenje, Microsoft PowerPoint, Videoigre u obrazovanju, Unity

Sadržaj

1. Uvod	1
2. E-učenje.....	2
2. 1. Definicija e-učenja	2
2.2. Povijest e-učenja	5
2.3. Prednosti i nedostaci e-učenja	7
3. Microsoft PowerPoint	10
3.1. O Microsoft PowerPointu.....	10
3.2. Prednosti i nedostaci korištenja PowerPointa	11
3.3. Alternative PowerPointu	13
4. Videoigre u obrazovanju	15
4.1. Korištenje videoigara u obrazovanju.....	15
4.2. Povijesni pregled videoigara u obrazovanju	17
4.3. Prednosti i nedostaci videoigara u obrazovanju	18
5. Unity.....	21
5.1. Unity igrin pokretač	21
5.2. Osnovni pojmovi unutar Unityja.....	22
5.3. Kreiranje projekta i korisničko sučelje Unityja.....	24
5.4. Unity tržnica imovinom	26
6. Podloga za kreiranje prezentacijske videoigre	27
6.1. Opis videoigre koja će se izrađivati	27
6.2. Unity 2D paket imovine	30
6.3. Imovina za izgradnju prostora.....	32
6.4. Ploče s obavijestima	36
6.5. Neutralni likovi	43
6.6. Prepreke i prelazak na sljedeću razinu	50
7. Kreiranje prezentacijske igre.....	55

7.1. Početni projekt i kreiranje nove scene.....	55
7.2. Kreiranje prostora i umetanje lika	56
7.3. Kreiranje ploča s obavijestima	62
7.4. Kreiranje neutralnih likova.....	67
7.5. Dodavanje prepreka te izrada igre	70
8. Korištenje videoigre u nastavi i moguća proširenja	73
8.1. Načini korištenja videoigre u nastavi	73
8.2. Usporedba s PowerPointom	74
8.3. Moguća proširenja.....	75
9. Zaključak.....	77
10. Literatura	79
11. Prilozi	82

1. Uvod

U današnjem obrazovanju upotreba računala je sve prisutnija. Pored toga, sve više su učitelji osviješteni o raznim alatima koji im stoje na raspolaganju kako bi mogli svoju nastavu učiniti zanimljivijom. Tako se učitelji obrazuju o raznim alatima kako bi mogli koristiti gotove sadržaje koje ti alati nude ili pak koristeći se tim alatima kreirati vlastiti sadržaj.

Primjer jednog alata koji se koristi za kreiranje vlastitih sadržaja za korištenje u nastavi je program PowerPoint Microsoft Office paketa. Koristeći se PowerPointom učitelji mogu lako kreirati vlastite prezentacije koje će koristiti kao potporu prilikom prezentiranja određenog znanja. Jednostavnost korištenja PowerPointa za kreiranje prezentacija je jedan od razloga za njegovu rasprostranjenost.

Pored PowerPointa i njemu sličnih alata za izradu prezentacija ima raznih alata za izradu sadržaja poput alata za izradu videa i zvučnih isječaka. Oni su nešto kompleksniji za korištenje te je iz tog razloga i njihova upotreba znatno rjeđa među učiteljima. Još jedna vrsta alata za izradu sadržaja koji stoji učiteljima na raspolaganju su alati za izradu videoigara. Međutim, izrada videoigara ljude često podsjeća na kompleksno programiranje te smatraju kako se bez znanja programiranja ne može kreirati videoigra. Samim time, alate za izradu videoigara smatraju previše kompleksnima te njihovo korištenje zaobilaze. Cilj ovog rada je pokazati kako se mogu kreirati videoigre bez znanja programiranja te kako ih se može koristiti kao potporu prilikom prezentiranja znanja. Uz to, ovaj rad će onima koji znaju programirati razjasniti kako mogu dalje proširiti temelj za takve videoigre kako bi izradu takvih videoigara učinili još fleksibilnijom.

2. E-učenje

Ideja korištenja videoigara u obrazovanju pripada konceptu e-učenja, u ovom poglavlju razradit će se sam koncept e-učenja. Razjasnit će se sama definicija e-učenja. Uza sam koncept e-učenja navode se i učenje na daljinu te mrežno učenje, razjasnit će se i definicija tih oblika učenja. Time će biti dana jasnija slika o samom e-učenju te će biti jasnije kako učitelji mogu uključiti videoigre u njega.

Pored toga, u ovom poglavlju će se razraditi povijesni pregled korištenja e-učenja u obrazovanju. Povijesni pregled će razraditi razvoj e-učenja u obrazovanju na razini svijeta te na razini same Republike Hrvatske. Također, pregled će iznijeti kako je uvođenje e-učenja utjecalo na učiteljev rad.

Na kraju poglavlja bit će iznijete prednosti i nedostaci koje e-učenje donosi. Pri tome, naglasak će biti na prednostima i nedostacima koje ono nosi za učitelje. Korištenje e-učenja u obrazovanju je posao na kojem učitelji uvijek moraju raditi te se za njega obrazovati, od velike je važnosti da ih se upozna s njegovim prednostima i nedostacima.

2. 1. Definicija e-učenja

Uz pojam e-učenja javljaju se i pojmovi poput mrežnog učenja i učenja na daljinu, te je zbog toga postavljanje jasne definicije za pojam e-učenja mnogim autorima bio veliki izazov. Iz tog razloga, ima mnogo definicija te postoje odstupanja među njima.

Pojam učenje na daljinu (engl. Distance learning) je najjednostavniji za pojasniti te kod njegove definicije su autori složni. Sherry (1996, str. 338-339) navodi: „Glavno obilježje učenja na daljinu je fizička i/ili vremenska odvojenost učitelja i učenika, mogućnost da učenik upravlja tokom učenja umjesto udaljenog učitelja te komunikacija između učenika i učitelja pomoću neke tehnologije.“

Phipps i Merisotis (1999) koriste malo drugačiju definiciju. Oni navode da se učenje na daljinu sastoji od sinkronog učenja i asinkronog učenja. Sinkrono učenje je učenje tijekom kojeg su učitelj i učenici fizički udaljeni, ali istovremeno komuniciraju. Dakle, udaljenost između učitelja i učenika stvara komponenta prostora. Asinkrono učenje je učenje tijekom kojeg učitelj i učenici nisu fizički udaljeni, ali se komunikacija između njih odvija u različito

vrijeme. Drugim riječima, u ovom slučaju udaljenost stvara komponenta vremena. Tomu još pridodaju kako se za komunikaciju koriste razni mediji za komunikaciju. Ukratko, prema njihovoj tvrdnji, učenje na daljinu je učenje tijekom kojega su učenik i učitelj odvojeni vremenom i/ili prostorom te za komunikaciju koriste multimedijske tehnologije.

Iz gore navedenih definicija se može zaključiti da je pojam učenja na daljinu jasno definiran te da označava oblik učenja tijekom kojeg su učitelj i učenik fizički odvojeni prostorom i/ili vremenom te se koriste raznim tehnologijama za komuniciranje.

Pored pojma učenja na daljinu javlja se i pojam mrežnog učenja (engl. Online learning). U starijim definicijama, pojam mrežnog učenja je bio iznimno sličan pojmu e-učenja. Richardson i Swan (2003) navode da se u početku mrežno učenje definiralo kao oblik predavanja odnosno učenja u kojem je dio sadržaja dostupan putem interneta, a učitelj i učenici nisu istovremeno povezani. . Prema navedenom, mrežno učenje je oblik predavanja u kojem je cijeli sadržaj dostupan putem interneta omogućavajući učenicima da sudjeluju bez obzira na njihovu geografsku lokaciju, neovisno o vremenu i prostoru.

Allen i Seaman (2007) kod definiranja mrežnog učenja navode četiri oblika učenja koji se razlikuju po količini sadržaja dostupnog na interneta. To su tradicionalno učenje koje nema sadržaja na internetu, učenje olakšano mrežom koje ima manje od 30% sadržaja na internetu, hibridno učenje koje ima između 30% i 80% sadržaja na internetu te mrežno učenje gdje je cijeli ili gotovo cijeli sadržaj dostupan na internetu i ne koristi se predavanje licem u lice.

Ukratko, iz gore navedenih definicija se može zaključiti kako je mrežno učenje oblik učenja u kojem je sadržaj dostupan putem interneta te nema nikakve komunikacije licem u lice između učitelja i učenika. Drugim riječima, mrežno učenje je oblik učenja na daljinu pri kojem se koriste mrežne tehnologije za komunikaciju.

Dobro bi bilo spomenuti i poseban oblik mrežnog učenja zvan MOOC. MOOC su masivni otvoreni mrežni tečajevi (engl. Massive Open Online Course). McAuley, Stewart, Siemens & Cormier (2010) navode da su masivni otvoreni mrežni tečajevi uglavnom besplatni te ih pohađa veliki broj učenika. Uz to, kod MOOC-ova, učitelj rijetko komunicira s učenicima. Rješavanje problema ili razjašnjavanje nejasnih dijelova učenici postižu komunikacijom s drugim učenicima koji pohađaju tečaj. Drugim riječima, MOOC-ovi su oblik mrežnog učenja gdje uglavnom nema obostrane komunikacije između učitelja i učenika, već učitelj

jednosmjerno pojašnjava sadržaj a potom učenici međusobno komuniciraju za rješavanje problema.

Nakon što su pojašnjeni pojmovi učenja na daljinu i mrežnog učenja moguće je jasnije definirati i sam pojam e-učenja bez miješanja s prethodno navedenim pojmovima. Jedna od novijih definicija e-učenja tvrdi da je „e-učenje oblik poučavanja čiji se sadržaj s učenicima dijeli putem digitalnog uređaja kao što su računalo ili mobitel te mu je namjena potpora pri učenju“ (Clark i Mayer, 2016, str. 8). Trinaest godina ranije, sličnu definiciju su dali i Welsh, Wanberg, Brown, & Simmering(2003) u kojoj tvrde da se e-učenje može definirati kao korištenje računalne mrežne tehnologije kako bi se dostavile instrukcije ili sadržaj pojedincima.

Iz ovih definicija se ne može uočiti razlika između e-učenja i učenja na daljinu ili mrežnog učenja. Iz njih se može zaključiti da su mrežno učenje i e-učenje gotovo jednak pojam, a učenje na daljinu je njihov krovni pojam.

Jochems, Koper, & Merriënboer (2004, str. 5) u svom radu navode: „integrirano e-učenje koje pokušava kombinirati elemente poučavanja licem u lice, učenja na daljinu i treninga na poslu.“ Pri tome, učenje na daljinu se odnosi na e-učenje kakvo je definirano u prethodno navedenim definicijama. Budući da se u obrazovanju upravo koristi takva kombinacija kod poučavanja, točnije bi bilo reći kako definicija e-učenja u obrazovanju odgovara definiciji integriranog e-učenja.

Liaw, Huang, & Chen, (2007) smatraju da e-učenje ustvari proširuje paradigmu tradicionalnog učenja kreirajući dinamične modele učenja tijekom kojeg se koriste računalne i mrežne tehnologije. Također tvrde da se u e-učenju aktivnosti učenja temelji na autonomnosti učenika i interaktivnom učenju te na mogućnosti surađujućeg oblika učenja gdje se suradnja odvija između učitelja i učenika ili između učenika samih.

Iz razloga što su mnoge definicije pojma e-učenja dosta nejasne Tavangarian, Leypold, Nölting, Röser, & Voigt, (2004) e-učenju pridaju vlastitu definiciju: E-učenje uključuje sve oblike elektronički podržanog učenja i poučavanja kojima je cilj izgradnja znanja s naglaskom na pojedinačno iskustvo, vježbe i znanje učenika. Pri tome, informacijski i komunikacijski sustavi, neovisno o tome jesu li umreženi ili ne, koriste se kao medij putem kojeg se implementira proces učenja.

Iz gore navedenih definicija može se zaključiti kako e-učenje nije samo oblik učenja u kojem se sadržaji dijele koristeći se računalnim i mrežnim tehnologijama. Ono je oblik učenja gdje se takve tehnologije koriste kao potpora i proširenje tradicionalnom obliku učenja. Upravo zbog toga, ono nije isto kao mrežno učenje jer se i dalje koristi tradicionalnim poučavanjem licem u lice samo što je ono prošireno koristeći se računalnom tehnologijom. Najbolji primjer toga je korištenje PowerPoint prezentacija u obrazovanju. Uz to, ne može se reći da je učenje na daljinu krovni pojam e-učenju budući da se dio e-učenja koristi poučavanjem licem u lice. Stoga, učenje na daljinu je širi pojam od e-učenja, no samo jedan dio e-učenja ulazi u učenje na daljinu.

Na kraju se može zaključiti kako je e-učenje korištenje računalnih i mrežnih tehnologija u svrhu kreiranja i dijeljenja nastavnih sadržaja, komunikacije između učitelja i učenika te učenika međusobno te proširivanja tradicionalnog oblika učenja. Pošto je cilj ovog rada predstaviti kako se mogu kreirati videoigre kao potpora u prezentiranju znanja, fokus ovog rada će biti na korištenju takvih tehnologija u svrhu kreiranja nastavnih sadržaja te proširivanja tradicionalnog oblika učenja.

2.2. Povijest e-učenja

Pojavu korištenja e-učenja i drugih prethodno navedenih oblika učenja je ponajviše uvjetovalo pojavljivanje novih tehnologija. Točnije rečeno, pojavu novih oblika e-učenja je omogućila velika dostupnost novih tehnologija. U slučaju da su određene tehnologije preskupe za svakodnevno korištenje u obrazovanju, novi oblici učenja se ne bi niti pojavili u javnom obrazovanju pošto bi tradicionalni oblici bili pristupačniji dok bi noviji oblici učenja bili dostupni samo bogatima. Samim time, razlika između bogatih i siromašnih bi postajala sve veća.

Međutim, kako nove tehnologije postaju sve pristupačnije time i novi oblici učenja postaju sve pristupačniji. Na taj način se i sama razlika između bogatijih i siromašnijih umanjuje. Time se postiže da su prilike koje učenici imaju podjednake te se korištenjem novih tehnologija u odgoju i obrazovanju omogućuje kako bi se postigla jednakost među učenicima.

Od prethodno definiranih pojmova, učenje na daljinu je oblik učenja koji je najviše uvjetovan širenjem novih tehnologija. Upravo su poboljšanja u poštanskoj usluzi u 19. stoljeću omogućila učenje na daljinu putem prepiske. Tait (2003) navodi kako se je prvi poznati

slučaj učenja na daljinu javio 1884. godine u Engleskoj kada je Isaac Pitman poučavao stenografiju putem prepiske.

Učenje na daljinu se odvijalo putem prepiske sve do širenja radija i televizije. Prema Sherry (1996) kod učenja na daljinu putem radija i televizije, učitelji su imali programe poučavanja koji su se emitirali diljem države. Međutim, problem takvog učenja je bio nedostatak dvosmjerne komunikacije. Problem nedostatka dvosmjerne komunikacije je bio riješen širenjem računalnih tehnologija pa su se tako počeli koristiti e-pošta i internet te se tako javilo mrežno učenje koje je od tada postao pretežito korišten oblik učenja na daljinu.

Paralelno s mrežnim učenjem se pojavilo i e-učenje te se dio mrežnog učenja kroz povijest preklapa s e-učenjem. „Naziv e-učenje prvi je put stvoren 1999. godine na seminaru Computer Based Training (CBT) o sustavima, ali je ideja o e-učenju bila već prije osmišljena“ (TalentLMS, baz dat.).

Ćukušić i Jadrić (2012) navode da se često smatra kako se e-učenje razvilo iz učenja na daljinu jer su zadržane i poboljšane osnovne ideje fleksibilnosti, modularnosti i skalabilnosti s pomoću informacijsko-komunikacijske tehnologije. Dalje navode da podrijetlo e-učenja se javlja već 1984. godine kad su se pojavili računalni tečajevi na disketama, koje su kasnije zamijenjene CD-ima. Kako razvoj tehnologije uvjetuje i razvoj e-učenja, sljedeći korak u povijesti učenja je bila pojava World Wide Weba 90-ih godina koji je zamijenio CD-e. U početku su se tiskani obrazovni materijali počeli dijeliti mrežno da se smanje troškovi tiskanja i transporta. World Wide Web je kreirao virtualno okruženje za učenje u kojima učenici i učitelji mogu indirektno komunicirati pomoću hardvera i softvera te je omogućio učenicima pristup širem spektru obrazovnih materijala naspram klasičnih knjižnica i dostave materijala poštom. Još jedna razvoj u tehnologiji koji je omogućio i razvoj e-učenja je pojava WEB 2.0 tehnologija koje su omogućile dinamičniji sadržaj i daljnje poboljšanje komunikacije između učenika i učitelja.

Razvoj e-učenja možemo promatrati kroz četiri razdoblja: obrazovanje u obliku treninga koji vodi instruktor, razdoblje multimedije, prvi val e-učenja, te drugi val e-učenja. Prvo razdoblje, koje je trajalo do 1983., je bilo prije široke dostupnosti računala. Tada je primarna metoda poučavanja polaznika bila u obliku organiziranog tečaja za računalom u fizičkom prostoru koji je vodio učitelj. Ova metoda je bila iznimno skupa te je trošila puno vremena.

Razdoblje multimedije je trajalo od 1984. do 1993. koje se pojavilo uz diskete i CD-e. Ovo razdoblje karakterizirano je računalnim tečajevima dijeljenim na disketama ili CD-ima. Time se povećala dostupnost obrazovanja no nedostajala je interakcija s učiteljem i raznolikost sadržaja.

Razdoblje prvog vala e-učenja traje od 1994. do 1999. U tom se razdoblju koristi e-pošta, internetski preglednici za pregledavanje mrežnog sadržaja te programi za pregledavanje raznih audio i video sadržaja. U drugom valu e-učenja, koji traje od 2000. do danas, zbog tehnoloških promjena u razvoju interneta i internetskih medija te pojava WEB 2.0, mijenja se organizacija i provođenje procesa učenja i poučavanja. Takve promjene su utjecale na promjene u učenju te su omogućile veću uštedu i višu kvalitetu iskustva učenja. (Ćukušić i Jadrić, 2012)

2.3. Prednosti i nedostaci e-učenja

E-učenje kao i svaki drugi oblik učenja sa sobom nosi određene prednosti i nedostatke. Ti prednosti i nedostaci mogu se odnositi na učenike kao krajnje korisnike e-učenja i na učitelje kao stvaratelje sadržaja koji se koristi u sklopu e-učenja.

Arkorful i Abaidoo (2015) daju opširan pregled prednosti i nedostataka koji dolaze s korištenjem e-učenja u obrazovanju. Analizom drugih radova o e-učenju navode kako je najčešće navedena prednost e-učenja mogućnost fokusiranja na potrebe individualnih učenika. Potom navode sljedećih sedam prednosti:

1. Fleksibilnije vrijeme učenja zbog mogućnosti odabira kada i gdje će se učiti.
2. Efikasniji prijenos znanja zbog velike dostupnosti informacija.
3. Lakša komunikacija s učiteljem te između samih učenika korištenjem mrežnih tehnologija.
4. Ušteda novaca zbog manje potrebe za putovanjem.
5. E-učenje uvažava razlike u preferiranim načinima učenja kod učenika.
6. E-učenje nadoknađuje nedostatak osoblja.
7. Učenici mogu odrediti tempo kod e-učenja te tako smanjiti frustriranost kod učenja.

S druge strane, kod analize nedostataka e-učenja najčešće navedeni potpuni, ili u slučaju škola djelomični, nedostatak interakcije licem u lice između učenika i učitelja te učenika međusobno. Pored toga navode sljedećih osam nedostataka:

1. Kod e-učenja postoji nedostatak interakcije zbog čega je još više nužna motivacija i upravljanje vremenom.
2. Objašnjavanje, razjašnjavanje i interpretiranje je manje efektivno kod e-učenja nego kod tradicionalnog učenja.
3. Zbog nedostatka interakcije licem u lice, moguće je da se pogorša interakcija između samih učenika.
4. Kod e-učenja, teže je nadzirati varanje.
5. Zbog neprimjerenog korištenja kopiranja i lijepljenja može doći do problema kao što su plagiranje i piratstvo.
6. Kod e-učenja, učenikove socijalne vještine se slabije razvijaju.
7. Neke discipline su manje pogodne za korištenje e-učenja.
8. Kod e-učenja može doći do neočekivanih troškova zbog zatrpanog Internet prometa.

Zhang, Zhao, Zhou i Nunamaker Jr. (2004) daju tablični prikaz koje prednosti i nedostatke nosi tradicionalni oblik učenja, a koje prednosti i nedostatke donosi e-učenje.

	Tradicionalno učenje	E-učenje
Prednosti	<ul style="list-style-type: none"> • Neposredna povratna informacija • Poznato i učitelju i učenicima • Motivira učenike • Njeguje društvenu zajednicu 	<ul style="list-style-type: none"> • Koncentrirano na učenika i ima mogućnost samoupravljanja • Ekonomičnije za učenike • Potencijalno dostupno svima diljem svijeta • Neograničen pristup znanju • Mogućnost arhiviranja za

		ponovnu upotrebu znanja i dijeljenje istog
Nedostatci	<ul style="list-style-type: none"> • Koncentrirano na učitelja • Vremenska i prostorna ograničenost • Skuplje 	<ul style="list-style-type: none"> • Nedostatak neposredne povratne informacije kod asinkronog učenja • Povećano vrijeme pripreme za učitelja • Neki ljudi se ne osjećaju ugodno kada ga koriste • Potencijalno više stresa, frustracije i zbunjenosti

Tablica 1. Prednosti i nedostati tradicionalnog učenja i e-učenja
Preuzeto iz (Zhang i drugi, 2004, str. 76)

Jedan od nedostataka koji se navodi je povećano vrijeme pripreme za učitelja. Dobro bi bilo napomenuti kako se kroz kreiranje videoigara na način predstavljen u ovom radu vrijeme pripreme neće umanjiti. Naprotiv, potrebno vrijeme za kreiranje videoigre, u usporedbi s vremenom potrebnim za kreiranje obične PowerPoint prezentacije, bi moglo biti još veće. Međutim, cilj ovog rada nije riješiti problem potrebno vremena za pripremu unutar e-učenja. Cilj ovog rada je predstaviti način na koji se videoigre mogu kreirati bez programerskog znanja budući da mnogi smatraju kako bez znanja programiranja ne mogu sami kreirati videoigre. S druge strane, potencijalan stres je još jedan od navedenih nedostataka. Kroz zabavne elemente videoigara on se može smanjiti.

3. Microsoft PowerPoint

Pošto je cilj ovog rada predstaviti način kako kreirati videoigre kao potporu u prezentiranju znanja, potrebno je ukratko predstaviti Microsoft PowerPoint koji je trenutno prevladavajući alat za izgradnju prezentacija koje se koriste prilikom prezentiranja znanja. Stoga će se u ovom poglavlju ukratko predstaviti sam PowerPoint, njegova povijest te prednosti i nedostaci koji dolaze s korištenjem PowerPointa.

3.1. O Microsoft PowerPointu

PowerPoint je prezentacijski softver Microsoftova Office paketa. PowerPoint koristi vizualni pristup prezentacijama koje u obliku prikaza slajdova podupiru oralno predavanje neke teme. Njegova upotreba je rasprostranjena, kako u poslovnom svijetu, tako i u obrazovanju. Upravo zbog njegove jednostavnosti i mogućnosti da bilo tko napravi prezentaciju koja izgleda profesionalno, Microsoft PowerPoint je „najkorišteniji alat za kreiranje prezentacija diljem svijeta“ (Russel, 2017). Uz to, Farkas (2008) navodi da PowerPoint prevladava u raznim sferama života kao što su industrija, vojska i obrazovanje.

PowerPoint kroz korištenje natuknica, grafova, slika, video i audio zapisa pomaže obogatiti usmeno izlaganje prezentacije. Koristeći se raznim medijima, osoba koja prezentira može poduprijeti ono što govori te vizualnim prikazom učiniti prezentirano znanje jasnije onome koji sluša prezentaciju. Pored toga, koristeći se PowerPointom, onaj koji prezentira može se voditi prema prezentaciji i tako osigurati pozorno praćenje predavanja. Uz to, koristeći se natuknicama u prezentaciji, može se podsjetiti što treba reći u danom trenutku.

Microsoft PowerPoint je napravljen za Windows i Macintosh operativne sustave. Iako je PowerPoint danas dio Microsoftova softvera, u početku PowerPoint je bio napravljen za Macintosh OS. Osmislila ga je tvrtka Forethought Inc. te se u to vrijeme program zvao Presenter. Prvu verziju programa su izdali u travnju 1987. godine i to pod imenom PowerPoint 1.0. U kolovozu iste godine, Microsoft je preuzeo Forethought Inc. Tek tri godine kasnije, 1990., izdana je prva verzija PowerPointa za Windows operativne sustave pod imenom PowerPoint 2.0. Pregled ostalih verzija programa se može vidjeti u tablici na sljedećoj stranici (fppt.com, 2016).

Windows			Macintosh		
Godina	Ime programa	Verzija	Godina	Ime programa	Verzija
1990	PowerPoint 2.0 for Windows 3.0	2.0	1987	PowerPoint 1	1.0
1992	PowerPoint 3.0 for Windows 3.1	3.0	1988	PowerPoint 2	2.0
1993	PowerPoint 4	4.0	1992	PowerPoint 3	3.0
1995	PowerPoint for Windows 95	7.0	1994	PowerPoint 4	4.0
1997	PowerPoint 97	8.0	1998	PowerPoint 98	8.0
1999	PowerPoint 2000	9.0	2000	PowerPoint 2001	9.0
2001	PowerPoint 2002	10.0	2001	PowerPoint X	10.0
2003	Office PowerPoint 2003	11.0	2004	PowerPoint 2004	11.0
2007	Office PowerPoint 2007	12.0	2008	PowerPoint 2008	12.0
2010	PowerPoint 2010	14.0	2010	PowerPoint 2011	14.0
2013	PowerPoint 2013	15.0	2015	PowerPoint 2016	15.0
2015	PowerPoint 2016	16.0			

Tablica 2. Pregled verzija PowerPointa

3.2. Prednosti i nedostaci korištenja PowerPointa

Pošto je cilj rada predstaviti videoigre kao alternativno rješenje PowerPointu prilikom prezentiranja znanja, dobro bi bilo upoznati se s prednostima i nedostacima koji dolaze s korištenjem PowerPointa, odnosno programa za kreiranja prezentacija na temelju prikaza slajdova.

Xingeng i Jianxiang (2012) u svome radu daju pregled prednosti i nedostatke koji dolaze s korištenjem PowerPointa prilikom poučavanja. Na kraju rada daju i prijedloge kako umanjiti navedene nedostatke.

Jedna od prednosti je *stvaranje boljeg vizualnog učinka te dublji dojam kod prezentiranja*. Kada je sadržaj predavanja apstraktan, moguće je koristiti se raznim vizualnim prikazima da bude jasniji. Druga prednost PowerPointa je ta što *PowerPoint omogućuje brži prijenos informacija*. Pošto kod PowerPointa učitelj ne mora pisati na ploču, više vremena može odvojiti za samo predavanje. Jednako tako, pošto se PowerPoint može lako dijeliti, učenici također ne moraju trošiti vrijeme na prepisivanje. Još jedna od prednosti je *veća točnost i sistematičnost predavanja*. Budući da se PowerPoint priprema unaprijed, prepravljanjem grešaka sadržaj može biti točniji te organiziranjem poretka slajdova sadržaj može biti sistematičniji (Xingeng i Jianxiang, 2012).

S druge strane, kod korištenja PowerPointa može doći do *korištenja nepotrebnih informacija*. Ponekad razne informacije, slike i animacije ubačene u PowerPoint ustvari odvlače pažnju od bitnoga. Pored toga, kod poučavanja uz PowerPoint učitelj može više biti okrenut prema prezentaciji a ne prema učenicima. Još jedan od nedostataka koje navode je taj da se učenici mogu lako umoriti ako na prezentacijama ima previše teksta te ne mogu više pratiti predavanje. Uz to, pošto je redoslijed slajdova u prezentaciji unaprijed određen i donekle linearan, samo predavanje gubi na dinamičnosti i njegov tijek ovisi o redoslijedu slajdova (Xingeng i Jianxiang, 2012).

Kao rješenje za smanjivanje tih nedostataka navode korištenja i PowerPoint prezentacije i ploče ili nekih drugih alata tokom predavanja. Uz to, navode da je dobro čim više istražiti mogućnosti PowerPointa te tražiti povratnu informaciju od učenika o samoj prezentaciji kako bi je mogao učiniti što prikladnijom za učenike (Xingeng i Jianxiang, 2012).

Slične prednosti i nedostatke korištenja PowerPoint prezentacija navodi i Soffar (2017). Kao jednu od prednosti navodi upotrebu raznih multimedijских sadržaja kako bi se prezentaciju učinilo zanimljivijom i da učenici budu usredotočeniji na nju. Pomoću Master slajda je moguće i osigurati konzistentnost prezentacije. Također, učitelj pomoću natuknica se može podsjetiti što treba reći tokom predavanja. A pošto je PowerPoint prezentacije lako dijeliti, nema potrebe za prepisivanjem. Nasuprot tome, animacije i određene pozadine mogu odvući

pažnju učenika. Također, linearna narav PowerPointa ograničava učitelju organizaciju sadržaja što pojašnjavanje kompleksnijih sadržaja čini težim. I na samom kraju kao nedostatak PowerPointa navodi ovisnost o računalnoj opremi.

Mogućnost da se poveća pozornost slušatelja korištenjem raznih slika, animacija, audio i video isječaka u PowerPointu kao prednost te mogućnost da se pažnja odvuče pretjeranim korištenjem istih kao nedostatak navodi i Lee (2011). Međutim, jednu od prednosti koja prethodno nije navedena je ta da za korištenje PowerPointa za kreiranje prezentacija nije nužno formalno obrazovanje pošto je sam program prilično intuitivan i bilo tko može lako naučiti raditi prezentaciju u njemu. Naspram kreiranja videoigara, kreiranje prezentacije je daleko jednostavnije. Stoga je cilj ovog rada učiniti kreiranje videoigara znatno lakšim kako bi se potaknulo učitelje da i sami kreiraju videoigre za svoja predavanja te ih tako učine dinamičnijima. Također, pošto videoigre nisu linearnog karaktera, mogućnost kreiranja fleksibilnog toka predavanja je znatno veća kod videoigara nego kod PowerPoint prezentacija. Način na koji se može omogućiti nelinearan tok videoigre će biti pojašnjen u kasnijim poglavljima.

3.3. Alternative PowerPointu

Budući da je Microsoft PowerPoint samo jedan od alata za izradu prezentacija, dobro bi bilo predstaviti još neke alate za izradu prezentacija. Neke od alternativa PowerPointu su Impress (LibreOffice, 2017) prezentacijski program Libreoffice paketa i Impress (OpenOffice, 2017) prezentacijski program Apache OpenOffice paketa. Razlog zašto se oba programa zovu isto je taj što su se i LibreOffice i Apache OpenOffice razvili iz paketa OpenOffice koji je bio otvorenog koda. Oba navedena programa su besplatna za korištenje za razliku od Microsoft PowerPointa. Pored njih, Slides alat (Google, 2017) google.docs paketa je još jedan besplatan alat za kreiranje prezentacija. Za razliku od oba Impressa, Slides je baziran na webu te za njegovo korištenje je samo potrebno imati web preglednik.

Još jedna od alternativa PowerPointu je web 2.0 alat Prezi. (Prezi, 2017) Prezi su 2009 godine osnovali Peter Arvai, Péter Halácsy i Adam Somlai-Fischer. Do danas, Prezi ima preko 75 milijuna korisnika i napravljenih preko 260 milijuna prezi prezentacija. Za razliku od PowerPointa i Impressa, Prezi se ne služi prikazom slajdova u izradi prezentacije. Prezi ima jedno veliko platno po kojem korisnik može postavljati razne sadržaje. Potom, tokom prezentiranja, osoba koja prezentira se kreće po tom platnu što daje bolju mogućnost kreiranja

vizualne povezanosti unutar sadržaja i tako nadoknaditi nedostatak loše povezanosti sadržaja koji dolazi s PowerPointom i sličnim prezentacijskim alatima. Samim time, oni koji slušaju prezentaciju mogu i sami bolje povezati sadržaj. Prezi je besplatan za korištenje u obrazovne svrhe, no alat se plaća za korištenje u druge svrhe.

4. Videoigre u obrazovanju

Budući da ovaj rad prikazuje mogućnost kreiranja videoigre u svrhu potpore prilikom prezentiranja znanja, može se naslutiti da će se te igre najviše koristiti u obrazovanju. Stoga je potrebno prikazati koja je trenutna uloga videoigara u obrazovanju. U ovom poglavlju će se prikazati kako se koriste videoigre u obrazovanju te koja je definicija obrazovnih videoigara, potom će se prikazati povijesni pregled korištenja videoigara u obrazovanju te će na kraju biti iznesene prednosti i nedostaci korištenja videoigara u obrazovanju.

4.1. Korištenje videoigara u obrazovanju

Zbog svoje repetitivne naravi, videoigre u obrazovanju najčešće se koriste u svrhu vježbanja ili ponavljanja. Budući da se videoigre igraju iznova, učenici koji bi igrali videoigre bi tako puno puta uvježbavali neku vještinu ili ponavljali neko gradivo. Vezano uz videoigre u obrazovanju, često se javlja i pojam *edutainment* koji je tvorenica od engleskih riječi *educational* što znači obrazovana i *entertainment* što znači zabava. Drugim riječima pojam *edutainment* označava obrazovnu zabavu (Squire, 2003).

Squire (2003) navodi dva oblika kako se igre koriste u obrazovanju. Prvi oblik videoigara koje se koriste u obrazovanju naziva igre vježbanja i prakse (engl. Drill and practice games). U takvim igrama učenici ponavljaju istu radnju i tako uvježbavaju određenu vještinu ili usvajaju određeno znanje te se najčešće koristi akcijski žanr videoigara zbog svoje repetitivne naravi. Uz to, takve igre je lako uključiti u tradicionalan kurikulum pošto ih je moguće najviše primijeniti izvan učionice, odnosno u vrijeme kada učenik uči sam. Neki od primjera takvih igara koje navodi su Alga-Blaster, Reader Rabbit i Knowledge Munchers.



Slika 1. Reader Rabbit



Slika 2. Knowledge Munchers

Drugi oblik videoigara koje Squire (2003) navodi su simulacijske i strateške videoigre. Naspram ostalih videoigara, simulacijske videoigre nastoje što vjernije prikazati svijet odnosno neki sustav unutar svijeta. Nadalje, simulacijske igre se može podijeliti na simulacije s visokom vjernošću koje pokušavaju svaki dio sustava što stvarnije prikazati i simulacije s niskom vjernošću koje sustav pojednostavljaju poradi isticanja određenog djela. Izrada simulacija s visokom vjernošću je skupa te se koriste samo kad nema drugog izbora. Primjer tomu su simulacije za treniranje pilota. S druge strane, simulacije s niskom vjernošću služe za lakše razumijevanje sustava kada se izbacе smetnje koje odvlače od bitnih dijelova sustava. Uz to, one omogućuju učenicima mijenjanje raznih parametara unutar sustava te tako uočavaju promjene koje se dogode unutar istog.

Slično tvrdi i Annetta (2008). Prema njoj, pošto zadržavaju pozornost korisnika, igre se u obrazovanju koriste kako bi učenik bio više uključen u vrijeme kada uči sam. Takav način korištenja odgovara prethodno navedenim videoigramama vježbe i prakse. Uz to, navodi kako se igre koriste za shvaćanje kompleksnih sustava što odgovara prethodno navedenim simulacijskim igrama. Povrh toga obrazovne videoigre često zahtijevaju korištenje raznih vještina kao što su logičko i kritičko razmišljanje, pamćenje i rješavanje problema. Stoga se može naslutiti da se videoigre mogu koristiti i u svrhu razvijanja tih vještina neovisno o sadržaju koji se uči.

S druge strane, Gareau (2009) daje primjer kako se u obrazovanju mogu koristiti, ne samo videoigre, već i igre općenito. On tvrdi da kada se igre koriste u obrazovne svrhe one se mogu

koristiti da se pouči novi sadržaj ili ponovi stari, da se učenici bolje socijaliziraju ili da motiviraju učenika.

Iz prethodno navedenog može se zaključiti kako se videoigre u obrazovanju koriste u sljedeće svrhe:

- privlačenja učenikove pozornosti i motivacije za učenje,
- vježbanje određenih vještina,
- učenje novih sadržaja i ponavljanje istih,
- razumijevanje kompleksnih sustava kroz drugačiji pristup sustavu.

Videoigre čiji će se način kreiranja predstaviti u ovom radu će se moći koristiti u svrhu učenja novih sadržaja i ponavljanja istih te za privlačenje učenikove pozornosti i motivacije za učenje

4.2. Povijesni pregled videoigara u obrazovanju

Iako se industrija videoigara znatno izmijenila u zadnjih 40 godina, za korištenje videoigara u obrazovanju kroz povijest nema puno promjena osim onih koje dolaze sa samim razvojem videoigara i tehnologije pomoću koje ih se kreira. Jedna od istaknutih promjena kroz povijest njihova korištenja javljanje je nove svrhe korištenja videoigara u obrazovanju. Tako Squire (2003) tvrdi kako su se u počecima korištenja videoigara u obrazovanju, drugim riječima devedesetih godina prošlog stoljeća, videoigre koristile u svrhu vježbanja i prakse. Krajem devedesetih godina i početkom tisućitih, videoigre su se počele koristiti i u svrhu omogućavanja učenicima da na temelju drugačijeg pristupa nekom sustavu mogu bolje dokučiti sam sustav. Upravo kod takvih igara, drugim riječima simulacija, razvoj tehnologije pomoću koje se videoigre razvijaju je jedino što pridonosi promjeni. Što je tehnologija postajala naprednija, te igre su mogle vjernije imitirati taj sustav i time omogućiti učenicima da što bolje dokuče ono što uče. Izuzev tog trenutka u povijesti kada su se igre počele koristiti i u svrhu razumijevanja sadržaja koji se uči, izuzev razvoja videoigara općenito, videoigre u obrazovanju se nisu mnogo izmijenile kroz povijest.

Što se tiče istaknutih videoigara koje su se koristile u obrazovanju kroz povijest, bilo bi dobro spomenuti dvije igre. Prva od njih je Second Life. Second Life je igra virtualnog svijeta koja pokušava što realnije imitirati stvaran svijet. Razvio ju je Linden Lab 2003. godine te je igra 2007. godine imala preko 6 milijuna igrača. U samom obrazovanju, Second Life se koristio

kako bi učenicima proširio iskustvo učenja tako da unutar virtualnog svijeta mogu posjetiti bilo koju lokaciju koja replicira stvaran svijet (Linden Lab, 2017). Igra je besplatna, ali puno stvari unutar igre je potrebno prvo platiti kako bi se mogle koristiti. Ukratko, Second Life je simulacija koja pokušava što realnije i što opširnije simulirati stvarni svijet, tako da kroz njega učenici mogu paralelno s učenjem učiti i uvježbavati vještine potrebne za život u stvarnom svijetu.

Druga od istaknutih igara je Minecraft: Education edition. To je obrazovna verzija popularne videoigre građenja svijeta Minecraft. To je igra otvorenog svijeta koja potiče kreativnost, suradnju i rješavanje problema unutar okoline u kojoj je jedina granica učenikova mašta. Obrazovna verzija je izdana krajem 2016. godine dok je sama igra Minecraft izdana 2011. godine. Igra košta 5 dolara po korisniku po godini, što može odbiti od njenog korištenja (Mojang, 2017).

4.3. Prednosti i nedostaci videoigara u obrazovanju

Neke od prednosti korištenja videoigara u obrazovanju se mogu uočiti i u prethodno navedenim svrhama korištenja istih. Kao što Squire (2003) i Annetta (2008) tvrde, jedna od prednosti je privlačenje učenikove pozornosti te održavanje visoke razine motivacije. Korištenjem videoigara može se postići da učenik bude više motiviran za učenje sadržaja koji mu sam po sebi nije toliko interesantan. Druga prednost videoigara u obrazovanju je mogućnost da učenici iskuse određeni sustav kroz drugačiji pristup. Takva prednost se najviše očituje kod simulacijskih videoigara, a Squire (2003) pridodaje da je prednost takvih videoigara što omogućuje učenicima da uče o određenom sustavu na temelju iskustva u trenucima kada učenje u pravom sustavu nije moguće.

Prednosti koje dolaze s korištenjem videoigara ovise i o žanru videoigre. Tako naprimjer De Aguilera i Mendiz (2003) navode da se kroz arkadne i platformerske videoigre razvija psihomotorika, kroz igre slagalica se razvija logičko razmišljanje, a simulacijske igre mogu pomoći pri razvoju svih intelektualnih vještina. Povrh toga, navode da igre pridonose razvoju vještina neovisnih o samom sadržaju koji se uči preko njih. Tako se preko igara razvija logičko razmišljanje jer je potrebno osmisliti razne načine rješavanja problema te vještina promatranja jer ih je zbog velikog broja elemenata na ekranu potrebno što bolje razlikovati. Uz to se razvija i vještina snalaženja u prostoru pošto se unutar igre kreće unutar prostora. Na

kraju navode da se kroz igre razvija i vještina strateškog planiranja kod zahtjevnijih igara te donošenje odluka kod igara s bržim tempom.

Iako se u radovima koji istražuju prednosti videoigara u obrazovanju nedostatci koji dolaze s korištenjem istih ne spominju, te nedostatke se može izvući iz nedostataka koji dolaze i sa samim e-učenjem. Jedan od tih nedostataka je povećano vrijeme pripreme za učitelja. Kod videoigara u obrazovanju, ne samo da učitelj treba uložiti znatno više vremena za pripremu, već je mogućnost prilagodbe tih videoigara za nastavu znatno ograničena njegovim znanjem. Ukoliko učitelj ne zna programirati, znatno ograničen je na samo postojeće videoigre koje je netko drugi izradio. Stoga će se u daljnjim poglavljima predstaviti jedan od načina na koji bi učitelj mogao kreirati vlastite igre za korištenje u nastavi. Međutim, vrijeme pripreme će i dalje biti znatno veće nego kod tradicionalnih oblika učenja. Drugi od nedostataka je potencijalno odvlačenje pažnje učenika pretjeranim korištenjem elemenata u pozadini koji nisu vezani uz ono što učenici trebaju naučiti. Uz to, manja količina interakcije licem u lice bi moglo pridonijeti problemima u socijalnom razvoju. Na kraju, kao i kod e-učenja, videoigre nisu jednako pogodne za različite discipline tako da je potrebno prvo razviti plan kako bi se videoigre mogle koristiti unutar određene discipline.

Pošto je cilj rada predstaviti videoigre kao potporu u prezentiranju znanja te kao alternativu PowerPointu, dobro bi bilo spomenuti koje prednosti, a koji nedostatci naspram PowerPointa bi se mogli javiti u korištenju i kreiranju videoigara na način predstavljen u ovom radu. Prednost korištenja videoigara kao potpore u prezentiranju znanja nad PowerPoint prezentacijama je bolja mogućnost organiziranja sadržaja. Pošto je prostor unutar videoigara znatno fleksibilniji za dinamično kretanje naspram prezentacija koje su linearne naravi, prednost videoigara je što se bolje mogu prikazati kompleksniji sadržaji te tijekom predavanja je prilagodljiviji. Nedostatak kod korištenja videoigara u prezentiranju naspram PowerPoint prezentacija je kompleksniji način kretanja kroz videoigru. Učitelj se tokom predavanja uz PowerPoint prezentaciju pritiskom na jedan gumb lako može kretati naprijed ili nazad kroz svoje predavanje. S druge strane, kod korištenja videoigara, učitelj mora upravljati likom da se kreće kroz igru kako bi se mogao kretati naprijed ili nazad kroz svoje predavanje. Samim time, učitelj je ograničen da se manje slobodno kreće kroz učionicu tokom predavanja te je vrijeme kretanja kroz videoigru potencijalno duže nego vrijeme kretanja kroz prezentaciju što bi moglo usporiti samo predavanje.

Na kraju se može zaključiti kako bi korištenjem videoigara u obrazovanju učitelji mogli potaknuti razvoj raznih vještina, kao što su logičko i kritičko razmišljanje, pored usvajanja sadržaja koje učenici trebaju naučiti. Međutim, za korištenje videoigara učitelj mora uložiti znatno više vremena za pripremu za nastavu kako bi te igre bile djelotvorne i da ne odvlače pažnju ili troše vrijeme predavanja.

5. Unity

U ovom poglavlju predstaviti će se okruženje unutar kojeg će se izgrađivati videoigra za prezentiranje znanja. Kako bi iduće poglavlje u kojem će se pojašnjavati kako se izgrađuju dijelovi potrebni za izgradnju videoigre bilo jasnije, u ovom poglavlju će se pojasniti što je to Unity igrin pokretač, te sam pojam igrinog pokretača. Pored toga, u ovom poglavlju pojasnit će se sučelje Unityja te određeni pojmovi i postupci koji su vezani uz Unity, a koristit će se kasnije u radu. Na kraju će se predstaviti Unity tržnica imovinom koja je važna za izgradnju videoigre bez programiranja.

5.1. Unity igrin pokretač

Za kreiranje videoigara predstavljene u ovom radu koristit će se Unity igrin pokretač stoga je potrebno upoznati se s Unityjem te sa samim pojmom igrinog pokretača. „Igrin pokretač (engl. Game engine) je softver koji pruža osnovne funkcije koje se redovito koriste za izgradnju videoigre“ (Okita, 2015, str. 5). Neke od tih funkcija su prikazivanje 2D i 3D oblika napravljenih pomoću softvera s treće strane kao što su Blender, 3D Studio Max ili Maya. Još jedna funkcija igrinog pokretača je mogućnost pisanja programskog koda kojeg će igrin pokretač koristiti (Okita, 2015). Dakle Unity je program koji omogućuje izradu videoigara.

Unity je izdan 2005. godine i izdala ga je tvrtka Unity Technologies. Posljednja izdana verzija Unityja je 5.6.0 koja je izdana 31. ožujka 2017. godine. Iako je za profesionalno korištenje Unityja potrebno imati plaćenu verziju, ukoliko je godišnja zarada kreatora igre ispod 100.000 dolara ili se Unity koristi u obrazovne ili druge nekomercijalne svrhe, onda se može koristiti besplatna verzija programa. Unity podržava programske jezike C#, Javascript i Boo. C# je najčešće korišten jezik u Unityju, te će sav kod u ovom radu biti napisan koristeći se C# jezikom. Također, dobro je navesti da će svi primjeri u radu biti izgrađeni u 5.6 verziji Unityja (Unity technologies, 2017).

Uz to, potrebno je napomenuti da je u početku Unity bio namijenjen za izradu 3D videoigara, pokretač se nazivao Unity 3D. U trenutku kada je napravljena podrška za 2D, igrin pokretač se počinje nazivati samo Unity iako se još uvijek mogu pronaći slučajevi gdje ga se naziva Unity 3D. To je problem neodređene terminologije, i nije greška rabiti bilo koji od naziva.

Kako je tip videoigara koji će se izrađivati u radu 2D videoigra, bilo bi neobično koristiti „3D“ dio imena te će se u daljnjem tekstu program navoditi samo kao Unity.

Prije no što se nastavi dalje s radom, bilo bi dobro predstaviti i neke alternative Unity igrinom pokretaču. Prva od alternativa i najveći suparniku Unityju je igrin pokretač Unreal Engine 4. Unreal Engine 4 je jako sličan Unityju i po mogućnostima izrade videoigara i po načinu plaćanja. Unreal Engine 4 također ima besplatnu inačicu koja se može koristiti u nekomercijalne svrhe. Međutim, kod u Unreal Engineu se isključivo piše u C++ programskom jeziku koji je nešto kompliciraniji od C# jezika te je samim time i izrada videoigara u njemu nešto teža.

Druge alternative Unity igrinom pokretaču koje bi se mogle koristiti za izradu videoigara s namjenom da se koriste prilikom prezentiranja znanja su Game Maker Studio 2 (YoYo games, 2017) i RPG Maker MV (Enterbrain Inc, 2017). Oba igrina pokretača su znatno jednostavnija od Unityja i Unreal Engine igrinih pokretača te je pomoću njih znatno lakše kreirati videoigru bez da osoba koja izrađuje igru zna programirati. Međutim, nedostatak ovih pokretača nad Unityjem je taj što se oba moraju plaćati da bi ih se moglo koristiti.

Upravo zato što je Unity jednostavniji igrin pokretač za izradu videoigara od Unreal Enginea te je besplatan naspram Game Maker Studia 2 i RPG Makera MV, u ovom radu će se koristiti Unity za izradu videoigara.

5.2. Osnovni pojmovi unutar Unityja

Kako bi poglavlje u kojem se pojašnjava podloga za kreiranje videoigre bila jasnija, potrebno je objasniti nekoliko osnovnih pojmova unutar Unityja koji će se koristiti kasnije u radu. Prvi i najosnovniji od tih pojmova je objekt.

Objekt (engl. Object) je osnovni element unutar igre. Sve što se pojavljuje unutar igre je objekt, te se igra izgrađuje od objekata. Objekt se pak sastoji od komponenata (engl. Component) koje mogu biti slika, programska skripta, animator, Collider i slično. Pored toga, postoje dječji objekti koji su objekti unutar drugog objekta i roditeljski objekti koji su objekti unutar kojih se dječji objekt nalazi. Uz objekt, jedan od temeljnih pojmova je okvir (engl. Frame). Okvir je slika koja se učita u jednom trenutku unutar igre. Najčešće se unutar igre pokazuje 30 ili 60 okvira u sekundi.

Slike, odnosno modeli objekta se prikazuju pomoću komponente zvane Sprite Renderer. On dalje omogućuje da se uređuje sama slika ili da se postavi njen poredak na sloju. Poredak na sloju (engl. Order in layer) omogućuje da se odredi koja slika će se prikazati ispred, a koja iza. Slike čiji je poredak na sloju veći će se prikazivati ispred.

Komponente programske skripte se odnosi na kod koji su programeri napisali, a koji je zadužen za upravljanjem objektima unutar igre. Sve programske skripte imaju dvije temeljne metode Start i Update. Metoda Start se poziva kada se pokrene igra, dok metoda Update se poziva na svakom okviru i sav kod unutar te metode je zadužen za upravljanje igrom nakon što se ona pokrene.

Animator (engl. Animator) je komponenta koja je zadužena za pokretanje i zaustavljanje određenih animacija objekta. On omogućuje objektu da izgleda živo.

RigidBody (engl. RigidBody) je komponenta koja objektu omogućuje da se ponaša kao fizičko tijelo te omogućuje fizici Unityja da upravlja tim tijelom. Uz RigidBody, često se javlja i komponenta Collider.

Collider (engl. Collider) je komponenta koja omogućuje da igra detektira kada se dva objekta dodiruju. Ukoliko neki objekt koji na sebi ima komponentu RigidBody i Collider naiđe na drugi objekt koje na sebi ima Collider, taj prvi objekt neće moći prolaziti kroz drugi. Ukoliko je potrebno da prvi objekt može prolaziti kroz drugi, ali da se opet dodir između njih može detektirati, potrebno je na Collider drugog objekta označi polje IsTrigger kao istinito.

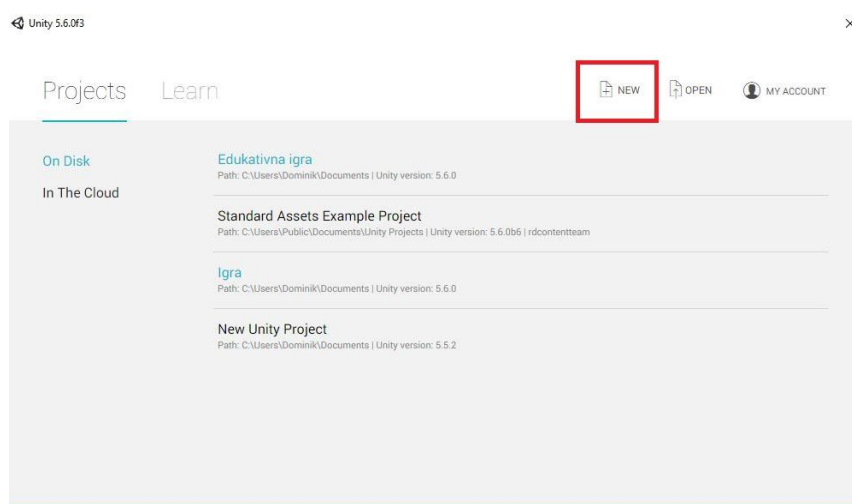
Svaka videoigra kreirana u Unityju ima svoju imovinu (engl. Assets). Imovina igre su programske skripte, 2D i 3D modeli, animatori, zvukovi, slike i predlošci te se od njih izgrađuje funkcionalna videoigra. Drugim riječima, imovina videoigre je skup svih gradivnih elementa videoigre unutar Unityja.

Predlošci (engl. Prefabs) su unaprijed izrađeni objekti koji su potom pohranjeni u imovinu te su spremni za ponovno korištenje. Pri tome, gotovo se ništa na njima ne treba mijenjati ili su te promjene iznimno male. Koristeći se izrađenim predlošcima, netko tko ne zna programirati može kreirati funkcionalnu videoigru.

Scena (engl. Scene) je dio igre koji predstavlja određenu cjelinu videoigre. Na scenu se postavljaju objekti kako bi se kreirala jedna cjelina, a potom se od skupa scena izgrađuje cijela igra.

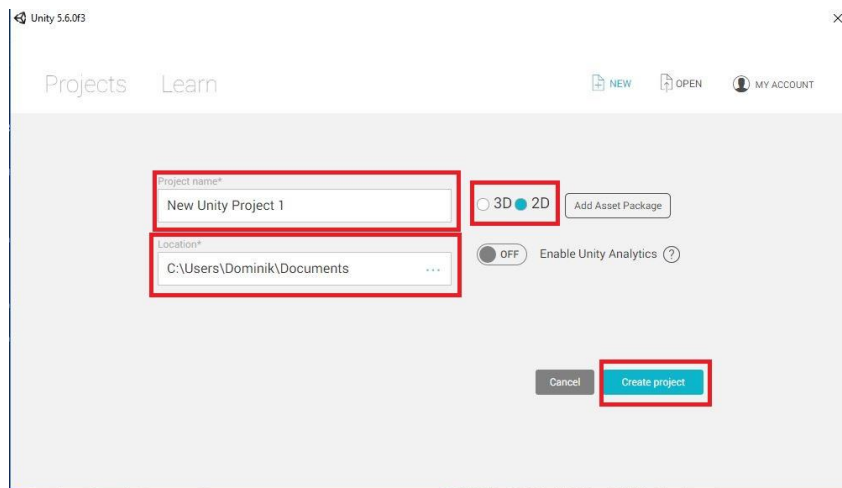
5.3. Kreiranje projekta i korisničko sučelje Unityja

Kod kreiranja videoigre u Unityju, ponajprije je potrebno kreirati projekt unutar kojeg će se izgraditi igra. Započeti projekt se može tako da se u početnom prozoru Unityja klikne na gumb New.



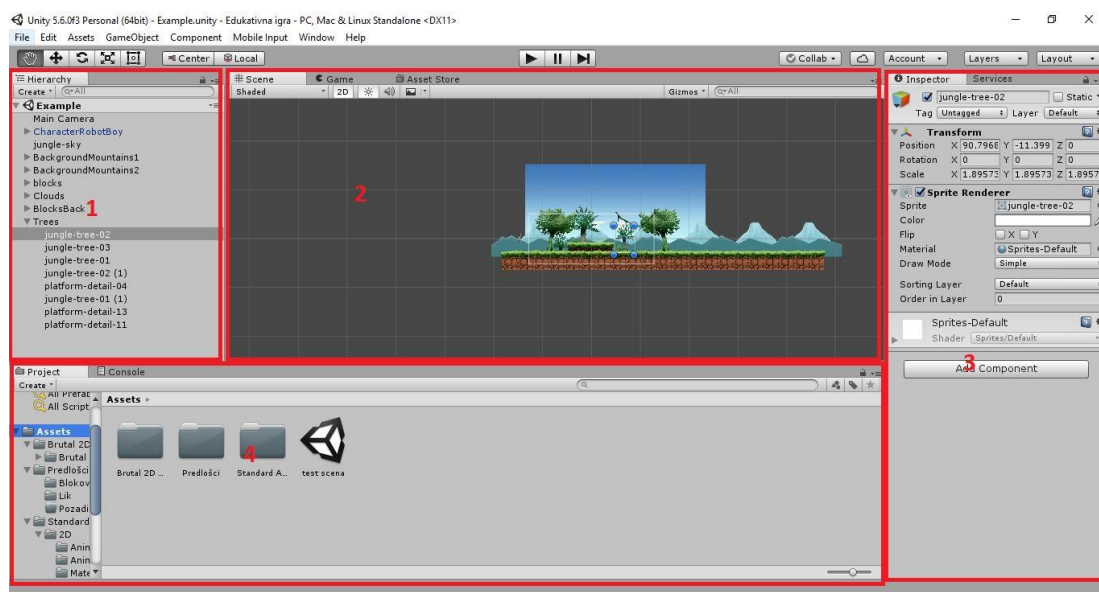
Slika 3. Početni prozor Unityja

Potom se otvara sučelje za kreiranje novog projekta gdje se biraju osnovne postavke projekta. U prozoru za kreiranje novog projekta bira se ime samog projekta, lokacija gdje će projekt biti pohranjen te hoće li projekt biti u 2D-u ili 3D-u. U ovom slučaju se bira 2D, što će podesiti kameru tako da odgovara našim potrebama. Postavke 2D-a ili 3D-a mogu biti naknadno izmijenjene unutar projekta. Na kraju se klikne na gumb Create Project kako bi se kreirao projekt.



Slika 4. Biranje postavka projekta

Nakon što se kreira projekt, otvara se sučelje Unityja čije je dijelove potrebno pojasniti kako bi se razumio kasniji dio u kojem se pojašnjava kreiranje podloge za videoigru. Sučelje se može vidjeti na slici 6.



Slika 5. Sučelje Unityja

U prozoru označenom s brojem 1 na slici 6. se nalazi hijerarhija (engl. Hierarchy). Unutar hijerarhije se mogu vidjeti svi objekti koji se nalaze na sceni te ih se preko nje može označiti, kopirati, brisati i pretraživati.

U prozoru označenom s brojem 2 na slici 6. nalaze se prozori scena (engl. Scene) i igra (engl. Game). Unutar prozora scena dodajemo objekte u igru te uređujemo kako će naša igra

izgledati. S druge strane, unutar prozora igra možemo vidjeti kako naša igra izgleda kada ju pokrenemo.

U prozoru označenom s brojem 3 na slici 6. nalazi se prozor inspektor (engl. Inspector) unutar kojeg možemo vidjeti komponente koje su na objektu te možemo podešavati podatke koji su dio tih komponenata.

U prozoru označenom s brojem 4 na slici 6. se nalaze prozori projekt (engl. Project) i konzola (engl. Console). Unutar prozora projekt se nalazi sva imovina igre te se iz nje mogu povlačiti predlošci na scenu kako bi se izgradila igra. U prozoru konzola se ispisuju pogreške unutar koda kada je igra pokrenuta.

Još jedan od dijelova sučelja koji bi trebalo pojasniti je Unity tržnica imovinom koja se nalazi u prozoru označenom s brojem 3 na slici 6. skupa s prozorima scena i igra.

5.4. Unity tržnica imovinom

Unity tržnica imovinom (engl. Unity Asset Store) je dio Unityja koji s jedne strane ljudima omogućuje da izrađuju i prodaju imovinu za videoigre, dok s druge strane ljudima omogućuje da kupuju imovinu te ju koriste u izradi vlastite igre. Dakle, jedna osoba može napisati određen kod za igru, modelirati lik, napraviti glazbu za videoigru te ju prodati na tržnici. Potom druga osoba može taj kod, lik, glazbu i drugo kupiti te pomoću njih izgraditi videoigru bez da ima znanja o programiranju, modeliranju ili skladanju. Unity tržnica imovinom je ta koja omogućuje ljudima da kreiraju videoigru bez da znaju programirati, modelirati ili bez znanja drugih vještina koje su potrebne za izradu videoigara. Kako se ne bi previše zalazilo u programiranje, određen dio koda će biti preuzet s Unity tržnice imovinom. Kod iz preuzete imovine će biti ukratko opisan unutar ovog poglavlja kako bi se imalo uvid u funkcionalnosti koje on omogućuje. Međutim, neće se pojašnjavati sam kod koji omogućuje te funkcionalnosti kako se ne bi previše zalazilo u programiranje pošto je cilj samog rada predstaviti način kreiranja videoigre bez poznavanja programiranja. Jednako tako, sve slike, zvukovi i likovi će biti preuzeti s tržnice imovinom.

6. Podloga za kreiranje prezentacijske videoigre

Nakon što je pojašnjena teorijska pozadina za korištenje videoigara kao potpore u prezentiranju znanja te predstavljeno okruženje unutar kojeg će se ona izrađivati, može se preći na izgradnju same igre. Ponajprije će biti objašnjeno kakvu videoigru će se izgrađivati kako bi se mogao dobiti kratak uvid u mogućnosti korištenja te videoigre. Nakon toga, gradivni elementi i kod koji su potrebni za izgradnju igre će biti pojašnjeni kako bi mogućnosti i ograničenja ovakve igre bile što jasnije te da onima koji imaju znanja o programiranju bude omogućeno dalje proširivati vlastitu igru.

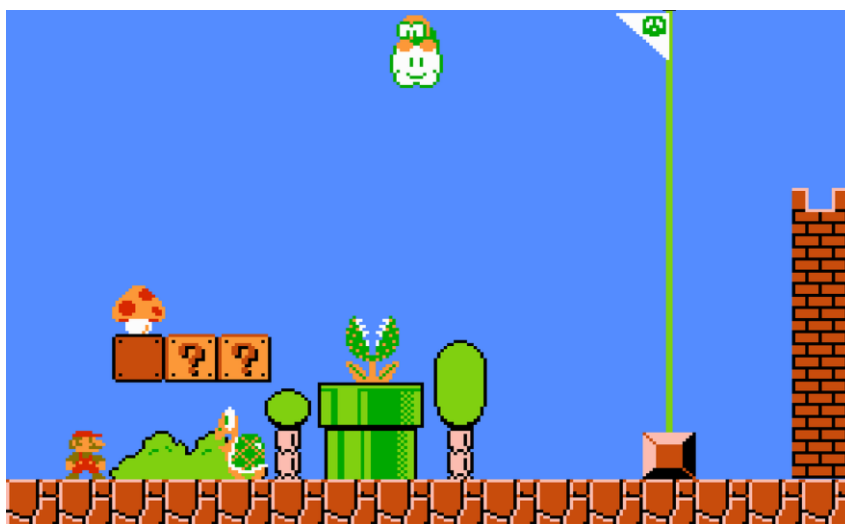
6.1. Opis videoigre koja će se izrađivati

Prije no što se krene s opisivanje pojedinih komponenti, odnosno imovine od koje će se izrađivati videoigra za prezentiranje znanja potrebno je opisati kakva će to igra biti. Na taj način će biti jasnije čemu pojedina imovina služi u videoigri, te će biti omogućen uvid kako se ta imovina uklapa u cjelinu videoigre. Pored toga, opisivanjem mogućnosti i ograničenja te videoigre te sam način na koji se ona može koristiti kao potpora prilikom prezentiranja znanja će biti jasniji ako se opiše sama igra.

Ponajprije, potrebno je odrediti kakav tip videoigre će se izrađivati u radu. Budući da je svrha igre koja će se izrađivati potpora u prezentiranju znanja, tip videoigre koji bi najviše odgovarao toj svrsi bi bio tip igara igranje uloga (engl. Role-playing game). Takav tip videoigara je bogat tekstom i sadržajem te najviše odgovara za prezentiranje određenog sadržaja, odnosno za poučavanje. Međutim, takav tip videoigara je znatno kompleksniji za izradu te se u ovom radu neće koristiti. Jedan od lakših tipova videoigara za izraditi je platformerski tip videoigre koji zahtjeva znatno manje programiranja naspram drugih tipova.

Također, za širu mogućnost organiziranja sadržaja, odnosno izbjegavanja linearnog slijeda u videoigri, 3D videoigre imaju znatno veću prednost nad 2D videoigramama jer se unutar 3D prostora povezanost između sadržaja može znatno bolje organizirati nego unutar 2D prostora. Međutim, kao i kod igara igranja uloga, 3D videoigre su kompliciranije za napraviti nego 2D igre.

Tip videoigre čije će se kreiranje opisati u ovom radu je 2D platformerska igra rađena u perspektivi iz profila. Neki od najpoznatijih primjera takvih igara su Super Mario, Sonic, Rayman, Braid, Metroid i Castlevania.



Slika 6. Super Mario

Cilj ovakvih igara je da igrač mora doći od točke A do točke B i pri tome izbjeći razne prepreke i protivnike skupljajući određene predmete na putu do cilja. Također, igrač na putu do cilja se može susresti s određenim neutralnim likovima (engl. Non-player character) s kojima može komunicirati kako bi dobio upute kamo treba ići ili kako se nositi s određenim preprekama. Osim neutralnih likova, igrač može naići i na ploče s obavijestima koje mu jednako tako mogu dati upute za kretanje dalje unutar igre.

Upravo ti neutralni likovi i ploče s obavijestima su temelj na kojemu će se raditi videoigra koja će se koristiti kao potpora u prezentiranju znanja. Ploče s obavijestima su prilično jednostavne za koristiti kod prenošenja sadržaja. Ploča na sebi može imati napisan određeni tekst ili imati na sebi neku sliku. Taj tekst ili slika predstavljaju sadržaj koji je potrebno prezentirati učenicima. Samim time, ploča s obavijestima funkcionira slično kao i sam slajd u

PowerPoint prezentaciji. Kao i slajd u PowerPoint prezentaciji, ploča može na sebi imati sadržaj koji učitelj treba prenijeti učenicima. Samim time, kretanjem od ploče do ploče te promatranjem njenog sadržaja može se postići jednak učinak prilikom prezentiranja kao i s PowerPoint prezentacijom. Učitelj se može kretati od ploče do ploče i tako prezentirati sadržaj koji ploče sadrže, kao što se kod PowerPointa kreće od slajda do slajda i tako prezentira sadržaj na slajdovima. Jedina razlika između ploča s obavijestima kod videoigre i slajdova kod PowerPoint prezentacija je mogućnost postavljanja ploča unutar prostora. Time se može bolje prikazati povezanost između sadržaja kojeg ploče prenose. Uz to, postavljanjem ploča unutar prostora može se omogućiti dinamičnost prezentacije i umanjiti linearnu prirodu prezentiranja koristeći se prelaženjem kroz slajdove u prezentaciji.

Nasuprot pločama s obavijestima koje imaju na sebi napisan neki tekst ili neku sliku kako bi prenijeli određeni sadržaj, neutralni likovi će se koristiti dijalogom kako bi prenijeli sadržaj. Drugim riječima, kada igrač dođe do neutralnog lika taj lik će mu dijalogom prenijeti određeni sadržaj. Samim time, neutralni likovi će funkcionirati drugačije nego slajdovi kod PowerPoint prezentacija. Za razliku od slajdova i ploča s obavijestima gdje učitelj detaljnije pojašnjava sadržaj koji oni prenose, kod dijaloga s likovima, prva od uloga učitelja je da on sam bude sudionik u dijalogu koji prenosi sadržaj. Na taj način učitelj se sam postavlja prema sadržaju koji mu lik prenosi. Time učitelj može pokazati učenicima kako se treba nositi s prezentiranim sadržajem. U današnje vrijeme, gdje je količina izvora iz kojih se mogu dobiti određene informacije sve veća, od iznimne je važnosti da učitelj pouči učenike kako se nositi sa sadržajem koji im je prezentiran. Time dijalog s neutralnim likovima učitelju omogućuje da učenicima pokaže kako se on nosi s određenim sadržajima i njihovim izvorima i svojim akcijama poučava učenike kako bi se i oni trebali postaviti prema istom. Nakon toga učitelj može nastaviti s detaljnijim objašnjavanjem sadržaja koji je predstavljen u dijalogu.

Osim toga što učitelj može pokazati učenicima kako se on nosi sa sadržajem, kada neutralni likovi predstavljaju sadržaj kroz dijalog, učenici taj sadržaj mogu povezati s tim likom i tako ga lakše upamtiti. Uz to, kao i kod ploča s obavijestima, neutralne likove se može postaviti unutar prostora i tako omogućiti dinamičniju prezentaciju i bolje ukazati na povezanost između sadržaja kojeg likovi prenose.

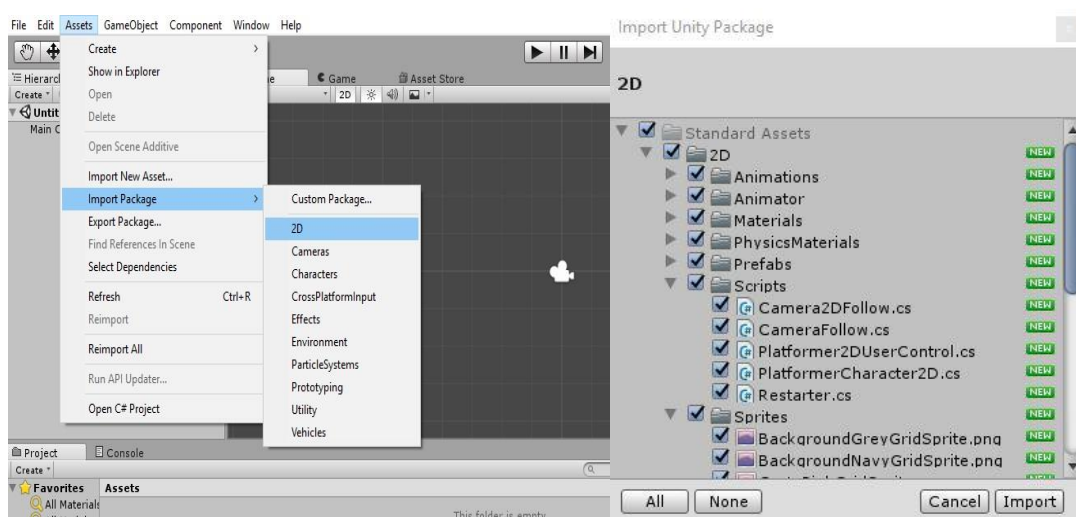
Ukratko, u ovom radu će se predstaviti način kako kreirati 2D platformersku videoigru u kojoj se igrač mora kretati od točke A do točke B i pri tom izbjegavati prepreke i protivnike.

Unutar prostora kroz koji se igrač mora kretati će se postaviti ploče s obavijestima te neutralni likovi koji igraču prenose određeni sadržaj te se na temelju tog sadržaja koji oni prenose može prezentirati znanje. Uz to, ovisno o tome na kojem položaju su ti likovi ili ploče postavljene može se pokazati povezanost između sadržaja koji prenose.

6.2. Unity 2D paket imovine

Prva od stvari koje su potrebne za funkcionalnu 2D platformersku igru je lik pomoću kojeg se možemo kretati unutar igre. Za izradu funkcionalnog lika potreban je 2D model, programska logika za njegovo kretanje te animator pomoću kojeg će model lika izgledati življe. Umjesto kreiranja svega navedenog od nule, iskoristiti će se Unity 2D paket imovine.

U prijašnjim verzijama Unityja, 2D paket imovine je bilo potrebno prvo preuzeti s tržnice imovinom da bi ga se moglo koristiti prilikom kreiranja videoigre. No u 5.6. verziji Unityja, taj paket dolazi u sklopu igrinog pokretača te ga je samo potrebno uvesti u projekt. To se radi tako da se unutar glavnog izbornika izabere Assets, potom Import Package i potom 2D. Nakon što se klikne na 2D paket, otvara se sučelje za odabir imovine unutar paketa koju želimo uvesti. Nakon što se odabere imovina koju treba uvesti klikne se na Import. Time je sva imovina dostupna za korištenje unutar projekta za izgradnju videoigre.



Slika 7. Uvezivanje 2D paketa imovine

Prvi i najvažniji dio paketa koji će se iskoristiti za izgradnju videoigre je sam lik. Unutar paketa, postoji model odnosno slike potrebne da se napravi 2D lik, animator koji upravlja

svim animacijama potrebnim za tečno kretanje lika, sve programske skripte koje omogućuju kretanje tog lika te njegov predložak (engl. Prefab). U predlošku su sve navedene komponente lika spojene te je lik spreman za korištenje.



Slika 8. Lik unutar 2D paketa

Što se tiče programskih skripta koje omogućuju kretanje lika, skripte Platformer Character 2D i Platformer 2D User Controller omogućuju kretanje lika unutar prostora. Skripta Platformer 2D User Controller nadzire igračev unos i na temelju tipki ili gumbova koje igrač pritisne prenosi skripti Platformer Character 2D te ona shodno tome izvršava odgovarajuću naredbu. Skripta Platformer Character 2D osigurava logiku koja je potrebna da se igrač kreće unutar prostora. Ona omogućuje da igrač pada prema dolje kad je pod utjecajem gravitacije te da prestane padati kada naiđe na tlo. Pored toga, ona omogućuje igraču da se kreće lijevo-desno te da može skakati ili puzati. Na kraju, skripta osigurava da igrač ne može prolaziti kroz objekte kroz koje on ne bi trebao moći proći. Ukratko, te dvije skripte omogućuju da igrač može upravljati likom te da se on može kretati kroz prostor koristeći hodanje, skakanje, puzanje i padanje.

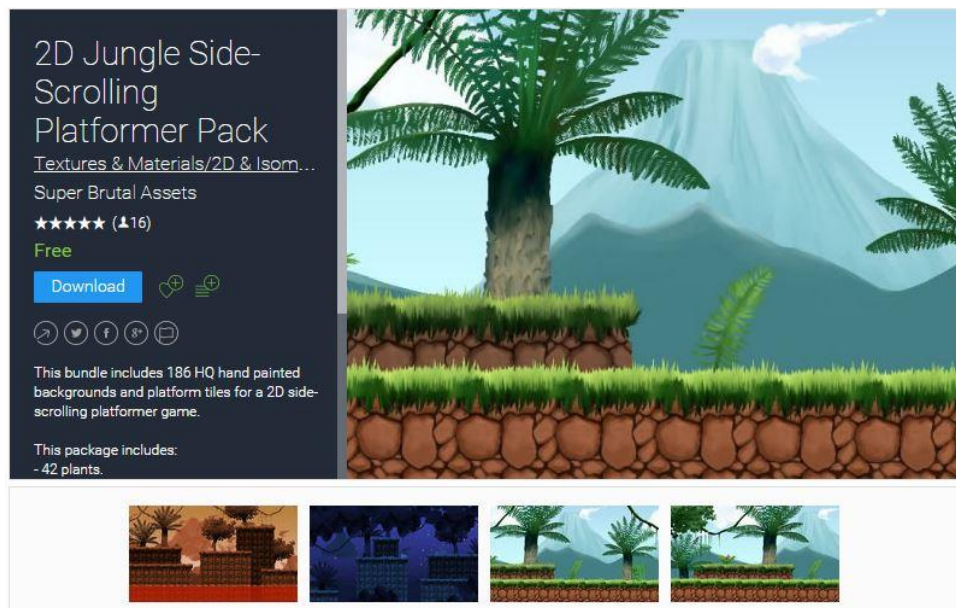
Iduća skripta koju će se koristiti iz 2D paketa je skripta Camera 2D Follow. Ova skripta omogućuje kameri igre da prati određeni objekt. Kako je bitno da kamera prati igračevog lika, objekt koji će kamera pratiti će biti sam lik. Time je omogućeno da se igrač kreće bilo kuda unutar prostora te da uvijek vidi na kojem mjestu se nalazi.

Posljednja od skripti unutar 2D paketa koju će se koristiti u izgradnji videoigre je skripta Restarter. Ta skripta omogućuje da se igra iznova pokrene kada igračev lik dotakne nešto što bi ga trebalo ubiti, bez obzira radi li se protivniku, zamci ili pak dnu igre.

Ukratko, Unity 2D paket imovine nam pruža model za 2D lika i animator koji kretnje tog lika čini življima. Pored toga, programske skripte unutar paketa omogućuju da se taj lik može kretati unutar 2D prostora. One još omogućuju da kamera prati lika dok se kreće kako bi igrač mogao vidjeti kuda se kreće, te omogućuje da određeni objekti mogu ubiti lika kad ih lik dotakne. Drugim riječima, ovaj paket imovine pruža sve dijelove koji su potrebni da bi se lik mogao funkcionalno kretati unutar igre, te da ga određene stvari mogu ubiti što je prvi dio koji je potreban za izgraditi 2D platformersku videoigru. Drugi dio potreban za izgraditi platformersku igru su same platforme, odnosno elementi od kojih će se izgraditi prostor dok treći dio koji je potreban su protivnici i prepreke. Pošto su elementi od kojih se izgrađuje prostor nužni za kreiranje videoigre koja će se koristiti za prezentiranje znanja, kako se izgrađuju predlošci tih elemenata te koje elemente će se koristiti u ovom primjeru će se pojasniti u sljedećem potpoglavlju. S druge strane, protivnici i prepreke nisu nužni za kreiranje videoigre za prezentiranje znanja, te će se njihova programska logika i način izgradnje predložaka pojasniti na kraju ovog poglavlja. Njih će se pojasniti tek nakon što se pojasni programska logika i način izgradnje predložaka ploča s obavijestima i neutralnih likova koji su potrebni za prenošenje znanja.

6.3. Imovina za izgradnju prostora

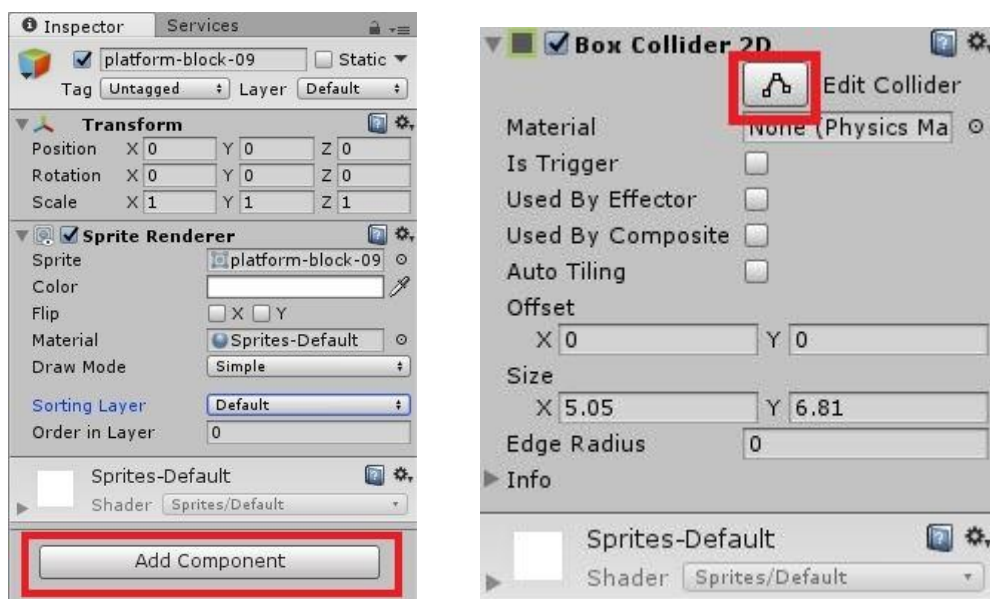
Za izgradnju prostora, potrebne su nam slike od kojih će se izgrađivati predlošci za gradivne elemente prostora. Jedan od paketa imovine koji sadrži slike potrebne za izgradnju igre je 2D Jungle Side-Scrolling Platformer Pack paket kojeg je napravio korisnik Super Brutal Assets. Taj paket sadrži 168 slika pozadine i pločica koje omogućuju izgradnju platformerske igre koja se odvija u džungli. Razlog zašto će se koristiti ovaj paket za izgradnju videoigre je što je taj paket besplatan za korištenje. Za preuzimanje tog paketa, potrebno je u tržnici imovinom kliknuti na gumb Download. Potom se otvara sučelje za odabir imovine koju želimo uvesti iz paketa, kao što je bilo kod Unity 2D paketa. Nakon što se paket uveze, od njega se mogu kreirati predlošci čiji će se postupak u nastavku opisati.



Slika 9.2D Jungle Side-Scrolling Platformer Pack

Iz slika unutar paketa potrebno je napraviti dva tipa predložaka gradivnih elemenata. Prvi tip gradivnih elemenata su blokovi kroz koje lik ne može prolaziti odnosno blokovi po kojima će lik hodati. Drugim riječima, to su blokovi od kojih će se izgraditi tlo po kojem će lik hodati, zidovi koji će blokirati liku put ili strop koji će ograničavati liku koliko visoko može otići. Drugi tip gradivnih elemenata su pozadinski blokovi i ukrasni blokovi. Svrha tih blokova je da igra izgleda ugodnije oku te nemaju svrhu da blokiraju igrača. Drugim riječima kroz te blokove bi igrač trebao moći nesmetano prolaziti.

Ponajprije će se opisati kako se rade predlošci za blokove kroz koje lik ne može prolaziti. Kako bi se izgradio predložak, ponajprije treba izvući sliku bloka iz prethodno navedenog paketa i postaviti je na scenu. Prilikom izrade predložaka, dobro je postaviti element iz kojeg se predložak radi na poziciju (0, 0) kako bi se izbjegle potencijalne greške oko postavljanja elemenata. Pošto bi blokovi kroz koje lik ne može prolaziti trebaju biti na istom sloju, njihov poredak na sloju će biti 0, kao i kod samog lika. Da bi se onemogućilo da lik može prolaziti kroz blokove, blokovima je potrebno dodati 2D Collider komponentu kako bi Platformer Character 2D skripta mogla detektirati kada lik naiđe na takav blok i tada onemogućiti liku da prođe kroz njega. Za dodavanje 2D Collider komponente, u inspektoru je potrebno kliknuti na gumb Add Component te odabrati valjani 2D Collider. Kao primjer, pokazat će se kako se dodaje Collider na blok koji je kvadratičastog oblika stoga odgovarajući Collider je Box Collider 2D.



Slika 10. Dodavanje komponente (lijevo), uređivanje Collidera (desno)

Nakon što je Collider komponenta dodana bloku, potrebno je podesiti koji točno dio slike će Collider pokrivati. Drugim riječima, potrebno je podesiti kroz koji dio slike lik neće prolaziti. Za podešavanje Collidera, potrebno je kliknuti na gumb za uređivanje Collidera kao što je pokazano na slici 11. desno. Nakon što se aktivira podešavanje Collidera, potrebno je samo mišem pomaknuti njegove stranice kako bi Collider pokrio odgovarajući dio slike. Za pomicanje stranice, potrebno je kliknuti na kvadratić na stranici i povući ju na odgovarajuću poziciju.



Slika 11. Stranice collidera

Nakon što je dodan Collider, blok je gotov te ga se može pretvoriti u predložak. Za kreiranje predloška blok se odvuče sa scene u sučelje s imovinom igre. Ukratko, za kreirati predložak

bloka kroz koji lik ne smije prolaziti, sliku se postavi na scenu, doda joj se odgovarajući Collider i podesi koji dio slike mora pokriti te na kraju blok iz hijerarhije postaviti u sučelje s imovinom igre.

Za kreiranje pozadinskih i ukrasnih blokova nije potrebno dodati Collider pošto lik treba slobodno prolaziti kroz njih. Jedino što treba podesiti kod njih je njihov poredak na sloju. Pozadinski blokovi trebaju imati manji poredak nego lik. Drugim riječima, njihov poredak na sloju mora biti negativni broj pošto je poredak na sloju lika 0. Što više pozadinski blok mora biti u pozadini, to manji mora biti njegov poredak na sloju. S druge strane, ukrasni blokovi koji moraju biti ispred drugih blokova moraju imati veći poredak na sloju naspram tih elemenata. Dakle, što više taj blok mora biti ispred drugih elemenata, tim veći mora biti njegov poredak na sloju. Za primjer pokazat će se kako se postavi poredak za pozadinski blok drveta koje se nalazi u bližoj pozadini. Stoga će njegov poredak na sloju biti negativan broj s malom vrijednosti. Kao i kod prethodno navedenih blokova, prvo se postavi slika na scenu. Potom se u Sprite Renderer komponenti u Inspectoru pod Order in Layer postavi određeni negativni broj kao što je pokazano na slici 13. Nakon toga, kao i kod prethodnih blokova, za kreiranje predloška potrebno je odvući blok iz hijerarhije u sučelje s imovinom igre.



Slika 12. Postavljanje poretka na sloju

Sljedeći prvi postupak za kreiranje predložaka blokova kroz koje se ne može prolaziti te sljedeći drugi postupak za kreiranje predložaka za pozadinske i ukrasne blokove, moguće je kreirati sve blokove koji su potrebni za izgradnju prostora videoigre. Samim time, dva nužna dijela za izgradnju platformerske 2D videoigre su pojašnjena te se može preći na kreiranje ploče s obavijestima i neutralnih likova koji prenose određeni sadržaj što je potrebno za kreiranje igre koja će se koristiti za prezentiranje znanja.

6.4. Ploče s obavijestima

Ploče s obavijestima funkcioniraju tako da igrač može unutar igre doći do ploče te ju potom aktivirati da se prikaže njen sadržaj. Ponajprije, kad igrač dođe do ploče, na ekranu se trebaju prikazati upute što treba pritisnuti da pročita sadržaj ploče. Kad pritisne odgovarajuću tipku, prikazuje se sadržaj koji je na ploči te se na ekranu ispisuju upute kako da igrač prestane čitati sadržaj ploče. Kako to izgleda, prikazano je na slikama ispod.



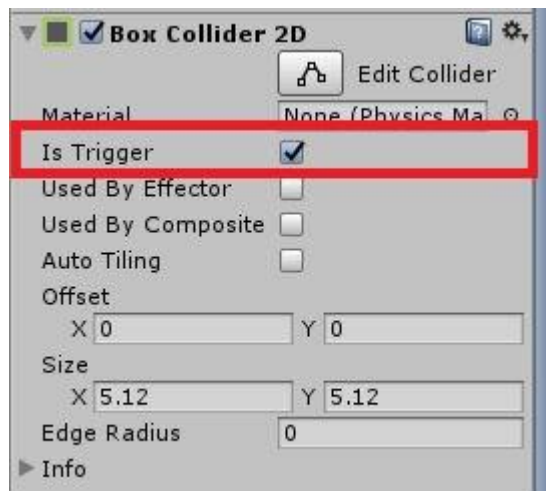
Slika 13. Ploča prije aktivacije (lijevo), ploča nakon aktivacije (desno)

Ovdje će se objasniti programska podloga za ploču s obavijestima te će se prikazati kako se izrađuje predložak za ploču. Kako se sam sadržaj dodaje na ploču će se pokazati na primjeru u sljedećem poglavlju.

Prvo što je potrebno za ploču s obavijestima je slika koja će predstavljati ploču u prostoru gdje igrač mora doći da bi mogao aktivirati čitanje sadržaja ploče. Pošto na tržištu imovinom nema odgovarajućih slika, pokazat će se postupak kako ubaciti običnu sliku u imovinu igre. Preporučljivo je koristiti slike u PNG formatu koje nemaju pozadinu te se na taj način bolje uklapaju u sam prostor igre. Za ubacivanje slike u imovinu igre, dovoljno je sliku odvući u Unityjevo sučelje s imovinom.

Nakon što imamo sliku u imovini, za izradu predloška prvo je potrebno postaviti sliku na scenu tako da ju se iz imovine odvuče na scenu te je su postavi na poziciju (0,0). Sljedeće što je potrebno za ploču s obavijestima jest dodati Collider kao što je prethodno pokazano kod imovine za izgradnju prostora. Jedina razlika kod Collidera između ploče s obavijestima i imovine za izgradnju prostora je što igrač mora biti u mogućnosti prolaziti kroz Collider ploče, a opet javljati igri da je igrač ušao ili izašao iz Collidera. To se radi tako da se u inspektoru na

komponenti Collider pod polje Is Trigger postavi istinita vrijednost kao što je prikazano na slici ispod.



Slika 14. Postavljanje polja Is Trigger u Collideru

Nakon što je u Collideru Is Trigger postavljeno na istinitu vrijednost, igrač može slobodno prolaziti kroz Collider ploče, a igra će pomoću metoda `OnTriggerEnter2D` i `OnTriggerExit2D` izvršavati odgovarajući kod. Da bi se izvršavao odgovarajući kod za ploču s obavijestima, potrebno je na sam objekt dodati i komponentu s odgovarajućom programskom skriptom. Za dodavanje komponente programske skripte, potrebno je pritisnuti gumb `Add Component` te upisati naziv odgovarajuće programske skripte. U ovom slučaju, to je skripta pod nazivom `Notice Board` čiji će se kod nadalje pojasniti. Na početku skripte se nalazi nekoliko polja koja su nužna za funkcioniranje koda.

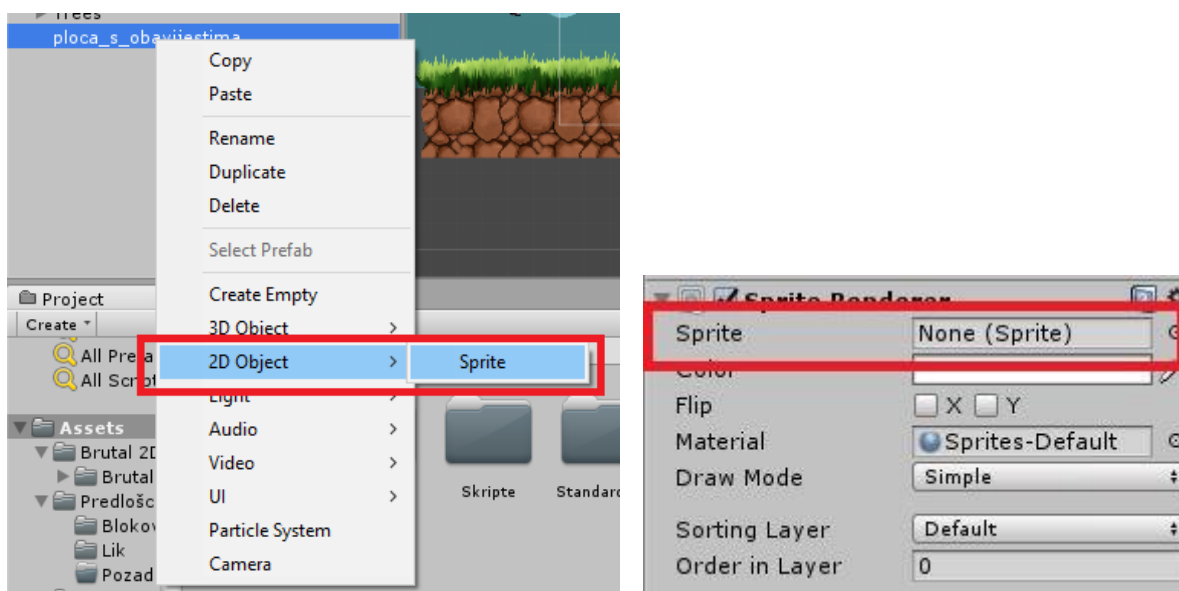
```
bool enteredBoundary;  
bool readingActive;  
public GameObject BoardBackground;  
public GUIStyle style;
```

Kod 1. Polja unutar skripte Notice Board

Polje `enteredBoundary` je tipa `boolean` te služi za provjeru je li igrač unutar Collidera ploče. Ono je postavljeno na istinito kada je igrač unutar Collidera, te je postavljeno na neistinito kada igrač izađe iz Collidera. Ono služi igri da provjeri kada mora igraču prikazati upute za aktivaciju ploče te kada mora igraču omogućiti da aktivira ploču. S druge strane, polje `readingActive` služi igri da provjeri da li igrač čita sadržaj ploče te samim time govori igri da prikaže upute za izlazak iz čitanja sadržaja i omogućuje igraču isto učini.

Polje BoardBackground je tipa GameObject što predstavlja objekte unutar Unityja. Dostupnost polja je postavljena kao public kako bi se to polje moglo vidjeti u inspektoru te mu se tako unutar istog može dodati odgovarajući objekt. BoardBackground polje treba sadržavati dječji objekt samog objekta ploče koji sadrži pozadinu sadržaja ploče. Također, taj objekt će u sebi imati dječje objekte koji će biti sadržaj ploče čije će se dodavanje pojasniti u idućem poglavlju.

Za dodavanje dječjeg objekta ploči za obavijest, potrebno je u hijerarhiji desnim klikom kliknuti na ploču s obavijestima, te pod 2D Object odabrati sprite. Nakon što je dodan dječji objekt tipa Sprite, u inspektoru na komponenti Sprite Renderer u polju Sprite je potrebno odabrati sliku koja će biti pozadina sadržaju ploče.



Slika 15. Dodavanje dječjeg objekta (lijevo), dodavanje Spritea (desno)

Prilikom povećavanja pozadine sadržaja ploče potrebno je postaviti da je veličina slike pozadine manja naspram kamere igre. Stoga je najbolje imati i kameru na sceni prilikom podešavanja. Na slici ispod, crvenom bojom je označena veličina kamere, dok je zelenom bojom označena veličina pozadine sadržaja ploče. Na kraju se pozicija pozadine postavi na (0,0) da bude u sredini objekta.

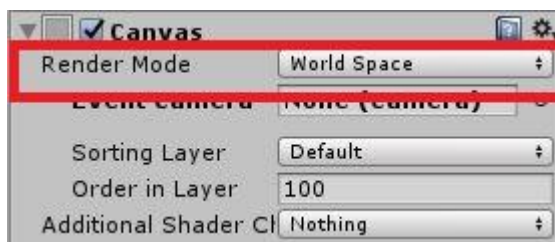


Slika 16. Omjer veličine kamere i pozadine sadržaja ploče

Da bi se taj dječji objekt dodao u polje BoardBackground potrebno ga je iz hijerarhije odvući na to polje unutar inspektora na komponenti Notice Board.

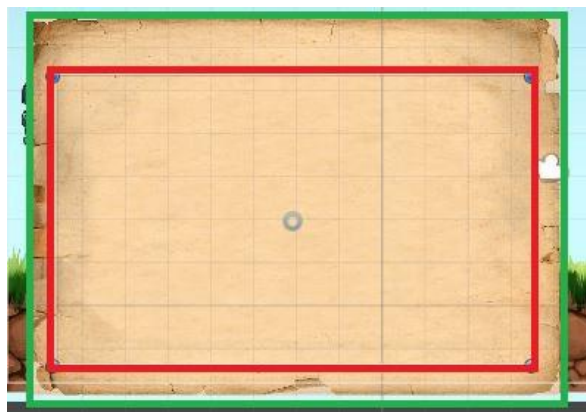
Samoj pozadini treba dodati još jedan dječji objekt koji će omogućiti prikaz teksta na pozadini, a to je objekt Canvas. Iako bi se Canvas trebalo moći dodati kao dječji objekt tako da se desnim klikom klikne na roditeljski objekt, u ovom slučaju na pozadinu, te se potom odabere pod UI pa Text. Canvas se u ovom slučaju ne dodaje kao dječji objekt nego kao zaseban objekt. Pri tome, u hijerarhiju se dodaje objekt Canvas koji kao dječji objekt ima Text te se uz njih dodaje i objekt pod imenom EventSystem koji se brine da se Canvas prikazuje. Dječji objekt Text se kod izrade predloška može obrisati, a objekti Canvas i EventSystem se označe te se odvedu u hijerarhiju na objekt pozadine kako bi ih se pretvorilo u dječji objekt.

Nakon toga je potrebno podesiti da veličina Canvasa odgovara veličini pozadine. Prvo što treba podesiti je da unutar komponente Canvas pod poljem Render Mode bude odabrana vrijednost World Space. Uz to, pod Order in Layer se postavi na 100 kako bi se sav tekst prikazao ispred pozadine.



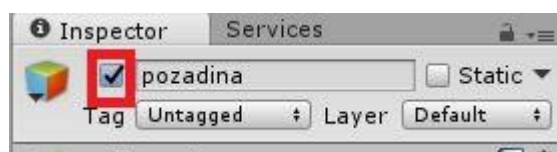
Slika 17. Render Mode Canvas komponente

Nakon toga, veličinu Canvasa je potrebno podesiti kao što se podešavala veličina pozadine naspram kamere. Za podešavanje veličine Canvasa pritisne se tipka **T** da se uđe u način rada za povećavanje objekata. Potom se povlačenjem rubova podesi veličina samog Canvasa.



Slika 18. Postavljanje veličine Canvasa (crveno) u odnosu na veličinu pozadine (zeleno)

Na kraju se dječji objekt pozadine postavi da nije aktivan jer za njegovo aktiviranje je odgovoran kod skripte. S druge strane, objekt Canvas i EventSystem se ostavljaju kao aktivni jer oni postaju neaktivni zajedno sa svojim roditeljskim objektom, pozadinom. Da bi se objekt postavilo kao neaktivan, potrebno je ukloniti kvačicu pored njegova imena u inspektoru kao što je pokazano na slici ispod.



Slika 19. Deaktiviranje objekta

Posljednje od navedenih polja je Style, koje je tipa GUIStyle te je postavljeno kao public kako bi se njegove vrijednosti mogle podešavati unutar inspektora. Svrha ovog polja je kako bi se odredio stil teksta kojim će se prikazivati upute za korištenje ploče. U ovom primjeru, font teksta je postavljen na Arial te je njegova veličina postavljena na 20. Uz to, pozadina teksta u normalnom stanju je postavljena na sivu te boja teksta na bijelu boju.

```
void Start () {
    enteredBoundary=false;
    readingActive=false;
    BoardBackground.SetActive(false);
}
```

Kod 2. Start metoda skripte Notice Board

U metodi Start, koja se poziva prilikom pokretanja igre, radi sigurnosnih mjera polja enteredBoundary i readingActive su postavljena na neistinita te dječji objekt BoardBackground je postavljen kao neaktivan.

```

private void OnTriggerEnter2D(Collider2D other)
{
    if (other.gameObject.tag == "Player")
    {
        enteredBoundary = true;
    }
}
private void OnTriggerExit2D(Collider2D other)
{
    if (other.gameObject.tag == "Player")
    {
        enteredBoundary = false;
        if (BoardBackground)
        {
            BoardBackground.SetActive(false);
            readingActive = false;
        }
    }
}

```

Kod 3. OnTriggerEnter2D i Exit2D metode skripte Notice Board

Metoda OnTriggerEnter2D je zadužena da kad igrač uđe u Collider ploče vrijednost polja enteredBoundary postavi na istinitu. Nakon toga, ostatak koda se pobrine da se igraču prikažu upute za čitanje sadržaja ploče te da isto može i učiniti. Metoda OnTriggerExit2D se pak brine da vrijednost polja enteredBoundary postavi na neistinitu što će dalje kodu omogućiti da ukloni upute s ekrana te da onemogući čitanje te ploče. Pored toga, ukoliko je igrač čitao sadržaj ploče, u ovoj metodi se dječji objekt ponovno ne prikazuje te polje readingActive bude neistinito ukoliko igrač izađe iz Collidera ploče.

```

void OnGUI()
{
    if (enteredBoundary && !readingActive)
    {
        GUI.Label(new Rect(Screen.width/2-135, Screen.height/15-10, 270, 20), "Pritisni Enter za čitanje!", style);
    }
    if (enteredBoundary && readingActive)
    {
        GUI.Label(new Rect(Screen.width / 2 - 135, Screen.height / 15-10, 270, 20), "Pritisni Escape za izlazak!", style);
    }
}

```

Kod 4. OnGUI metoda skripte Notice Board

Metoda OnGUI skripte NoticeBoard je zadužena za prikazivanje uputa igraču kako da čita sadržaj ploče ili kako da ugasi čitanje sadržaja. Ponajprije, ukoliko je polje enteredBoundary istinito, a polje readingActive neistinito, drugim riječima ako je igrač unutar Collidera ploče ali ju ne čita, prikazuju se upute kako pročitati sadržaj ploče. Tekst se prikazuje unutar oznake

pravokutnog oblika duljine 270 i visine 20 te pozicioniran na vrhu ekrana u njegovoj sredini. Kao sadržaj oznake je navedeno: „Pritisni Enter za čitanje!“, te je kao stil teksta oznake postavljeno polje Style. S druge strane, ako su polja enteredBoundary i readingActive oba istinita, odnosno igrač je unutar Collidera ploče te čita sadržaj, prikazana je ista oznaka, ali sa sadržajem „Pritisni Escape za izlazak!“

```
void Update () {
    if (enteredBoundary && !readingActive)
    {
        if (Input.GetKeyDown(KeyCode.Return) || Input.GetKeyDown(KeyCode.KeypadEnter))
        {
            readingActive = true;
            BoardBackground.SetActive(true);
        }
    }
    if (enteredBoundary && readingActive)
    {
        if (Input.GetKeyDown(KeyCode.Escape))
        {
            readingActive = false;
            BoardBackground.SetActive(false);
        }
    }
}
```

Kod 5. Update metoda skripte Notice Board

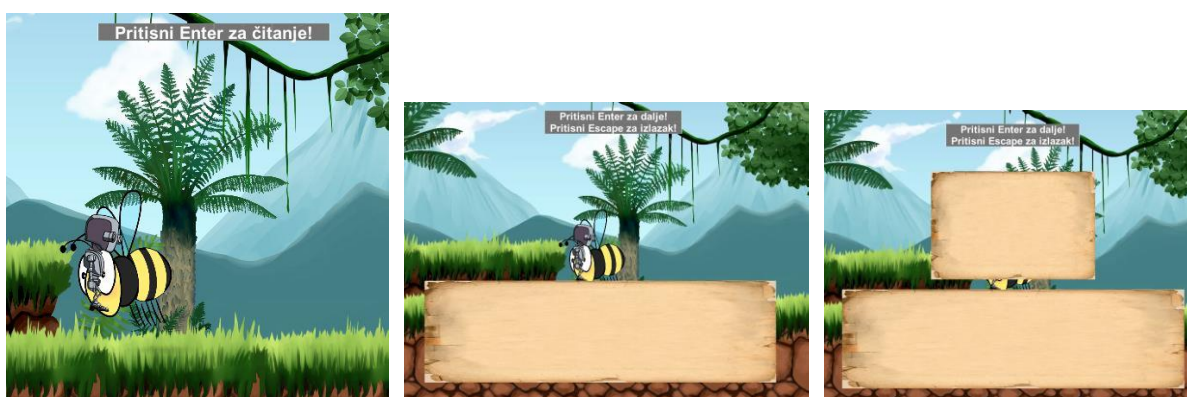
Update metoda programske skripte Notice Bord se brine da igrač može pročitati sadržaj ploče ili izaći iz čitanja istog. Ukoliko je polje enteredBoundary istinito i polje readingActive neistinito, tada kod provjerava unos igrača. Ukoliko igrač pritisne tipku Enter, polje readingActive se postavlja kao istinito i tako ostatku koda javlja da igrač čita sadržaj ploče. Potom se aktivira dječji objekt pozadine ploče te s njim svi dodani dječji objekti koji imaju u sebi sadržaj ploče kako bi ih igrač mogao pročitati. S druge strane, ako su polja enteredBoundary i readingActive istinita, kod provjerava kod igračevog unosa ako je pritisnuta tipka Escape. Ukoliko igrač pritisne tipku Escape, polje readingActive se postavlja kao neistinito te time se javlja kodu da je igrač prestao čitati sadržaj ploče. Nakon toga se dječji objekt pozadine ploče i svi njegovi dječji objekti s njim deaktiviraju i time se prekida čitanje sadržaja.

Ukratko, skripta Notice Board omogućuje da igrač kad uđe u Collider ploče može pritiskom na tipku Enter pročitati sadržaj ploče. Nakon toga, omogućuje igraču da pritiskom na tipku Escape, ili izlaskom iz Collidera ploče se prestane prikazivati sadržaj ploče. Nakon što su ploči s obavijestima dodani Collider kojem je polje Is Trigger postavljeno kao istinit,

programska skripta Notice Board te podešen dječji objekt pozadine sadržaja ploče, moguće je objekt spremi kao predložak. To se učini tako da se objekt sa scene odvuče u imovinu igre.

6.5. Neutralni likovi

Neutralni likovi funkcioniraju slično kao i ploče s obavijestima. Kad igrač dođe do neutralnog lika, na ekranu se pokazuju upute kako da započne razgovor s likom, a kad je u samom razgovoru, prikazuju se upute kako izaći iz razgovora ili nastaviti dalje s razgovorom. Za razliku od ploče s obavijestima koje pokazuju sadržaj unutar samo jednog dijela, neutralni likovi prikazuju sadržaj kroz razgovor koji se sastoji od nekoliko dijelova. Svaki dio se sastoji od teksta dijaloga te po potrebi može imati i popratnu sliku radi jasnijeg prikaza.



Slika 20. Neutralni lik prije aktivacije (lijevo), prikaz dijaloga bez slike (sredina), prikaz dijaloga sa slikom (desno)

Ovdje će se objasniti programska podloga potrebna da bi se mogao prikazati dijalog neutralnog lika te će se pokazati kako se izrađuje predložak neutralnog lika. Kako se sam tekst dijaloga i slike dodaju liku će biti pojašnjeno u sljedećem poglavlju.

Isto kao i kod ploče s obavijestima, prvo je potrebna slika neutralnog lika koji će se nalaziti u prostoru do kojeg igrač treba doći da bi započeo razgovor. Pošto u tržnici imovinom nema odgovarajuće imovine, slika za lika će se preuzeti s interneta te odvlačenjem u sučelje s imovinom se dodati u imovinu igre.

Prvi korak u kreiranju predloška neutralnog lika je isto kao i kod ploče s obavijestima da se slika lika odvuče na scenu i postavi na poziciju (0,0). Potom je potrebno dodati Collider i postaviti polje Is Trigger na istinito kako je pokazano kod ploča s obavijestima. Nakon što je Is Trigger postavljen kao istinit, igrač će moći nesmetano prolaziti kroz neutralni lik, a

programska skripta za njega će moći pokazivati odgovarajuće upute te omogućiti aktivaciju dijaloga.

Potom je neutralnom liku potrebno dodati odgovarajuću programsku skriptu kako bi dijalog mogao funkcionirati. Za funkcioniranje dijaloga neutralnog lika potrebne su skripta NPC, koja osigurava funkcionalnost dijaloga, te pomoćna skripta Has Picture Check, koja omogućuje provjeru ima li dio dijaloga i popratnu sliku.

```
public bool hasPicture;
```

Kod 6. Has Picture Check skripta

Has Picture Check skripta se sastoji samo od jednog polja, hasPicture. Polje hasPicture je tipa boolean te je postavljeno kao public kako bi se moglo uređivati u inspektoru. Ta pomoćna skripta se postavlja na same dijelove dijaloga, te se vrijednost polja hasPicture postavlja kao istinito ako taj dio dijaloga ima i popratnu sliku. Ova pomoćna skripta omogućuje skripti NPC da provjeri ima li dio dijaloga koji se prikazuje sliku, te ovisno o tome prikazuje i pozadinu za sliku. Naspram Notice Board skripte, skripta NPC ima nešto više polja.

```
bool enteredBoundary;  
bool readingActive;  
public GUIStyle style;  
public GameObject DialogueBackground;  
public GameObject PictureBackground;  
public GameObject[] Dialogue;  
private int NextDialogueIndex;
```

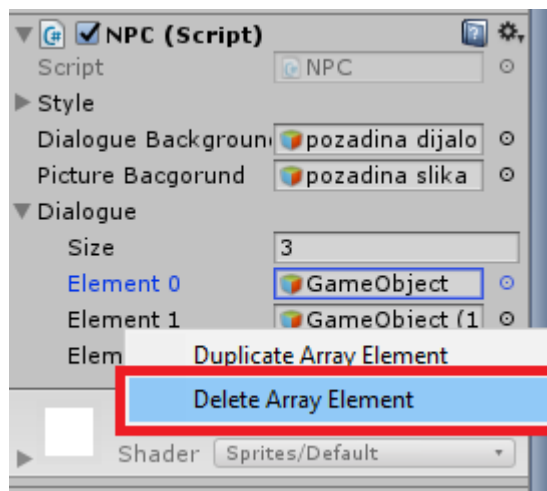
Kod 7. Polja skripte NPC

Polja enteredBoundary i readingActive su jednaka kao i u Notice Board skripti. Ona omogućuju kodu da provjeri ukoliko je igrač unutar Collidera neutralnog lika te čita li u tom trenutku dijalog lika ili ne. Polje style, jednako kao i kod Notice Board skripte, omogućuje podešavanje stila teksta uputa te se podešava na način kao što je pokazano kod ploča s obavijestima.

Polja DialogueBackground i PictureBackground su tipa GameObject, odnosno objekt unutar igre, te se ona odnose na dječje objekte unutar neutralnog lika koji predstavljaju pozadinu teksta dijaloga te pozadinu slike dijaloga. Kao i kod pozadine sadržaja ploče s obavijestima, veličina pozadine se podešava u odnosu na kameru igre. Za dodavanje tih objekata u odgovarajuća polja, potrebno ih je odvući iz hijerarhije u polje unutar komponente NPC u

inspektoru. Također, pozadini dijaloga je potrebno dodati i Canvas kao i kod ploče s obavijestima. Dakle, desnim klikom klikne se na pozadinu dijaloga te se pod UI odabere Text. Pošto će se Canvas staviti izvan objekta pozadine dijaloga, potrebno je Canvas i EventSystem odvući na objekt pozadine dijaloga da ih se postavi kao dječje objekte. Pored toga, dječji objekt unutar Canvas objekta se može obrisati.

Sljedeće od polja, polje Dialogue je lista koja prima tip GameObject, te se odnosi na objekte koji unutar sebe imaju dijelove dijaloga. Unutar te liste se nalaze svi objekti koji su dijelovi dijaloga koji se treba prikazati te je njihov poredak unutar liste jednak stvarnom poretku tih dijelova u dijalogu. Za dodavanje objekta u listu, potrebno je iz hijerarhije odvući objekt na odgovarajuće polje u inspektoru. Za promjenu redoslijeda elemenata unutar liste, potrebno je ili izmijeniti koji objekt je na kojem mjestu u listi, ili izbrisati objekt iz liste i postaviti ga na odgovarajuće mjesto. Za mijenjanje objekta na određenom mjestu u listi, potrebno je odvući novi objekt iz hijerarhije na njegovo mjesto u inspektoru. Za brisanje objekta iz liste, potrebno je desnim klikom stisnuti na taj objekt u listi te pritisnuti Delete Array Element. Kako se dodaju dijelovi dijaloga unutar objekta će biti pojašnjeno u idućem poglavlju.



Slika 21. Brisanje objekta iz liste

Posljednje od polja unutar skripte NPC je polje NextDialogueIndex koje je tipa integer te služi za provjeru koji se sljedeći dio dijaloga mora prikazati ili ako nema sljedećeg dijela dijaloga da se cijeli prikaz dijaloga ugasi.

```

void Start(){
    enteredBoundary = false;
    readingActive = false;
    DialogueBackground.SetActive(false);
    PictureBacgorund.SetActive(false);
    foreach (GameObject dialoguePart in Dialogue)
    {
        dialoguePart.SetActive(false);
    }
    NextDialogueIndex = 0;
}

```

Kod 8. Start metoda skripte NPC

Kao i kod skripte Notice Board, Start metoda skripte NPC služi za postavljanje vrijednosti polja radi sigurnosnih mjera. Vrijednosti polja enteredBoundary i readingActive se postavljaju kao neistinite te se objekti pozadine dijaloga i slike postavljaju kao neaktivni. Nadalje, pomoću petlje foreach prolazi se kroz svaki objekt koji je dio dijaloga te ga se postavlja kao neaktivnog. Na kraju, vrijednost polja NextDialogueIndex se postavlja na 0 kako bi se pri aktivaciji dijaloga prvo prikazao prvi dio dijaloga.

```

private void OnTriggerEnter2D(Collider2D other){
    if (other.gameObject.tag == "Player"){
        enteredBoundary = true;
    }
}
private void OnTriggerExit2D(Collider2D other){
    if (other.gameObject.tag == "Player"){
        enteredBoundary = false;
        if (DialogueBackground) {
            DialogueBackground.SetActive(false);
            PictureBacgorund.SetActive(false);
            Dialogue[NextDialogueIndex - 1].SetActive(false);
            readingActive = false;
            nextDialogueIndex=0;
        }
    }
}

```

Kod 9. OnTriggerEnter i Exit metode skripte NPC

OnTriggerEnter2D i OnTriggerExit2D metode su isto tako slične kao i kod Notice Board skripte. Enter metoda postavlja vrijednost polja enteredBoundary kao istinitu kako bi ostatak koda znao kada da prikaže upute za aktivaciju dijaloga te da omogući istu. Exit metoda

vrijednost tog polja stavlja kao neistinitu te deaktivira sve dijelove dijaloga ukoliko igrač izađe iz Collidera neutralnog lika. Jednako tako, vrijednost polja readingActive se stavlja kao neistinita kako bi ostatak koda znao da igrač više ne čita dijalog.

```
void OnGUI()
{
    if (enteredBoundary && !readingActive)
    {
        GUI.Label(new Rect(Screen.width / 2 - 100, Screen.height / 15 - 10, 270, 20), "Pritisni Enter za čitanje!", style);
    }
    if (enteredBoundary && readingActive)
    {
        GUI.Label(new Rect(Screen.width / 2 - 100, Screen.height / 15 - 10, 270, 45), "Pritisni Enter za dalje!\nPritisni Escape za izlazak!", style);
    }
}
```

Kod 10. OnGUI metoda skripte NPC

Metoda OnGUI je gotovo ista kao i kod skripte Notice Board. I ovdje se na ekranu ispisuje uputa sadržaja „Pritisni Enter za Čitanje!“ kad igrač uđe u Collider neutralnog lika a ne čita dijalog. S druge strane, kad je igrač unutar Collidera, no čita dijalog, prikazuje se uputa čitanje daljnjeg dijela dijaloga sadržaja „Pritisni Enter za dalje!“ te uputa za izlazak iz dijaloga sadržaja „Pritisni Escape za izlazak!“

```
void Update(){
    if (enteredBoundary && !readingActive)
    {
        if (Input.GetKeyDown(KeyCode.Return) || Input.GetKeyDown(KeyCode.KeypadEnter))
        {
            readingActive = true;
            DialogueBackground.SetActive(true);
            Dialogue[NextDialogueIndex].SetActive(true);
            var pictureChecker=
Dialogue[NextDialogueIndex].GetComponent<HasPictureCheck>();
            if (pictureChecker.hasPicture)
            {
                PictureBacgorund.SetActive(true);
            }
            NextDialogueIndex += 1;
        }
    }
}
```

```

else if (enteredBoundary && readingActive)
{
    if (NextDialogueIndex < Dialogue.Length)
    {
        if (Input.GetKeyDown(KeyCode.Return) || Input.GetKeyDown(KeyCode.KeypadEnter))
        {
            var Picture_Checker = Dialogue[NextDialogueIndex - 1].GetComponent<HasPictureCheck>();
            if (Picture_Checker.hasPicture)
            {
                PictureBacgorund.SetActive(false);
            }
            Dialogue[NextDialogueIndex - 1].SetActive(false);
            Picture_Checker = Dialogue[NextDialogueIndex].GetComponent<HasPictureCheck>();
            if (Picture_Checker.hasPicture)
            {
                PictureBacgorund.SetActive(true);
            }
            Dialogue[NextDialogueIndex].SetActive(true);
            NextDialogueIndex += 1;
        }
    }
}
else if (NextDialogueIndex >= Dialogue.Length)
{
    if (Input.GetKeyDown(KeyCode.Return) || Input.GetKeyDown(KeyCode.KeypadEnter))
    {
        readingActive = false;
        DialogueBackground.SetActive(false);
        Dialogue[NextDialogueIndex - 1].SetActive(false);
        var Picture_Checker = Dialogue[NextDialogueIndex - 1].GetComponent<HasPictureCheck>();
        if (Picture_Checker.hasPicture)
        {
            PictureBacgorund.SetActive(false);
        }
        NextDialogueIndex = 0;
    }
}
}
if (enteredBoundary && readingActive)
{
    if (Input.GetKeyDown(KeyCode.Escape))
    {
        readingActive = false;
        DialogueBackground.SetActive(false);
        Dialogue[NextDialogueIndex - 1].SetActive(false);
        var Picture_Checker = Dialogue[NextDialogueIndex - 1].GetComponent<HasPictureCheck>();
        if (Picture_Checker.hasPicture)
        {
            PictureBacgorund.SetActive(false);
        }
        NextDialogueIndex = 0;
    }
}
}

```

Kod 11. Update metoda skripte NPC

Na početku metode, ukoliko je vrijednost polja `enteredBoundary` istinita, a vrijednost polja `readingActive` neistinita, metoda provjerava unos igrača. Drugim riječima, ukoliko je igrač unutar `Collidera` neutralnog lika i ne čita njegov dijalog, kod provjerava ako je igrač pritisnuo tipku `Enter`. Ukoliko je tipka `Enter` pritisnuta, vrijednost polja `readingActive` je postavljena kao istinita da ostatak koda može prikazati odgovarajuće upute. Potom se pozadina teksta dijaloga postavlja kao aktivna te se prvi dio dijaloga postavlja kao aktivan. Potom se koristi pomoćno polje `PictureChecker` koje se odnosi na komponentu `Has Picture Check` preko koje se provjerava ima li prvi dio dijaloga i popratnu sliku. Ukoliko taj dio ima sliku, aktivira se i pozadina slike. Na kraju se polje `NextDialogueIndex` uvećava za 1 kako bi se pri idućem unosu prikazao sljedeći dio dijaloga.

Nakon toga, metoda provjera ako su vrijednosti polja `enteredBoundary` i `readingActive` oba istinita, drugim riječima ako igrač čita dijalog. Ako jesu, metoda provjerava ako je polje `NextDialogueIndex` manje od dužine liste dijelova dijaloga. Ako je, kod zna da ima još dijaloga za prikazati. Ukoliko igrač pritisne `Enter`, metoda prvo provjerava ima li prethodni dio dijaloga sliku. Ukoliko je ima, pozadina slike postavlja se kao neaktivna. Potom postavlja sam prethodni dio dijaloga kao neaktivan. Nakon toga, ako sljedeći dio dijaloga ima sliku, postavlja pozadinu slike kao aktivnu te aktivira sljedeći dio dijaloga. U protivnom samo aktivira sljedeći dio dijaloga. Na kraju, polje `NextDialogueIndex` uvećava za 1.

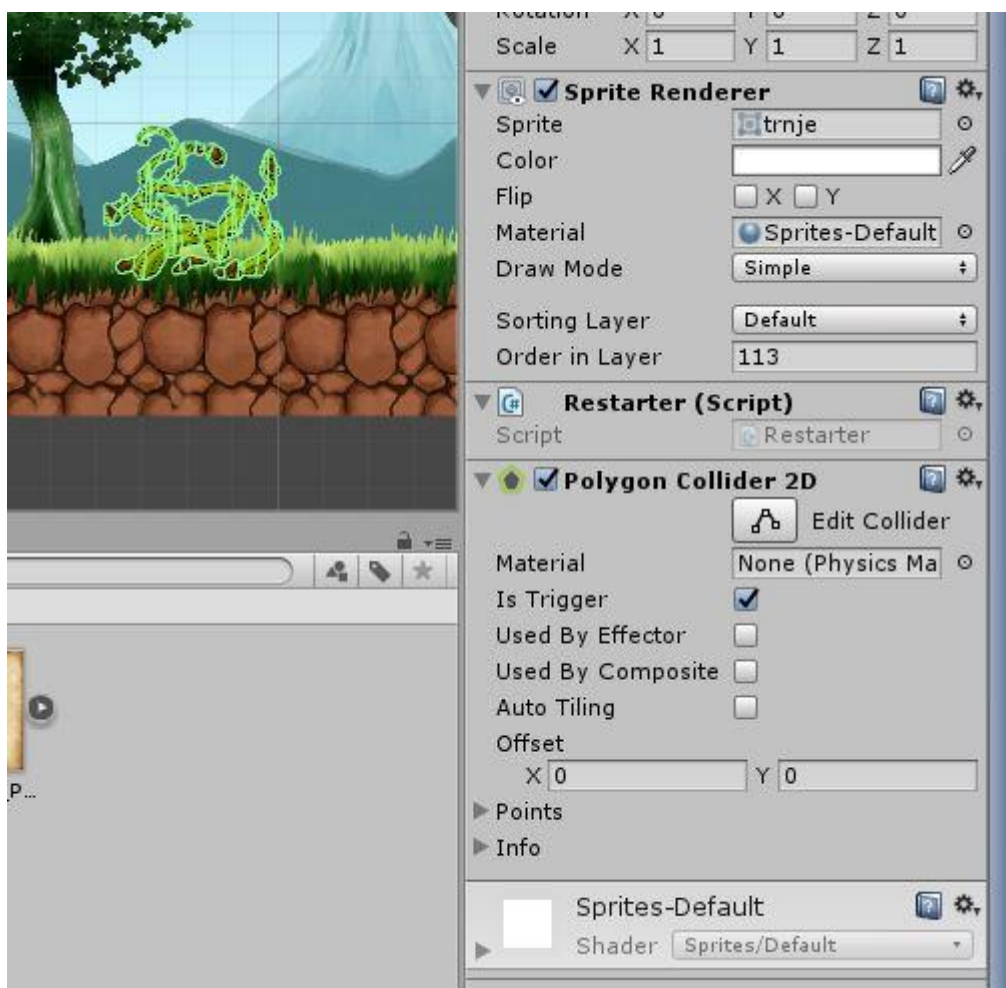
Nasuprot tome, sljedeći dio koda provjerava ako je `NextDialogueIndex` veći od dužine liste, odnosno ako više nema dijaloga za prikazati. Ukoliko je tako, kod postavlja pozadine kao neaktive, polje `readingActive` stavlja na neistinitu vrijednost te `NextDialogueIndex` vraća na 0. Jednako tako, na kraju `Update` metode se provjerava unos igrača dok čita dijalog. Ukoliko igrač pritisne tipku `Escape`, sve pozadine se deaktiviraju, te se polje `readingActive` stavlja kao neistinito i polje `NextDialogueIndex` na 0 kako bi se kasnije mogao iznova pokrenuti dijalog.

Ukratko, skripta NPC omogućuje igraču da kad dođe do neutralnog lika pritisne tipku `Enter` te započne dijalog. Tokom čitanja dijaloga, skripta omogućuje da igrač pritiskom na tipku `Enter` nastavi dalje kroz dijalog, ili pritiskom na tipku `Escape` izađe iz dijaloga. Nakon što su neutralnom liku dodani `Collider` kojem je polje `Is Trigger` postavljeno kao istinito, programska skripta `NPT` te podešen dječji objekti pozadine sadržaja teksta dijaloga i slike dijaloga, moguće je objekt spremi kao predložak. To se učini tako da se objekt sa scene odvuče u imovinu igre.

6.6. Prepreke i prelazak na sljedeću razinu

Kako bi igra bila malo zabavnija, potrebne su joj određene prepreke koje će ju učiniti više izazovnom. Međutim, cilj ovog rada je pokazati kako izgraditi igru kao potporu u prezentiranju znanja. Stoga će prepreke biti iznimno jednostavne te će biti ograničene na dva tipa. Prvi tip su stacionarne prepreke koje iznova pokreću razinu kada ih igrač dodirne. Drugi tip su pokretne zamke, koje jednako tako ponovno pokreću razinu kada ih igrač dotakne, no one će se unutar prostora i kretati.

Prvi korak u kreiranju predloška stacionarne prepreke je postaviti sliku prepreke na scenu i staviti ju na poziciju (0,0). Potom, za kreiranje predloška stacionarne prepreke koristit će se skripta Restarter iz Unity 2D paketa imovine. Ona omogućuje da se iznova pokrene razina na kojoj se trenutno igrač nalazi kada igrač uđe u Collider objekta koji ima tu skriptu. Uz skriptu je još potreban Collider koji će detektirati igračev ulazak u zamku. Pošto igrač prvo mora ući u taj Collider da bi se skripta pokrenula, potrebno je u inspektoru na komponenti Collider staviti da je vrijednost polja Is Trigger istinita.



Slika 22. Primjer stacionarne zamke

Primjer kako izgleda zamka u sceni te koje su njene komponente se može vidjeti na prethodnoj slici. U ovom slučaju je korišten Polygon Collider 2D jer preciznije prekrije sliku zamke od Box Collidera. Nakon što su zamci dodane potrebne komponente, dovoljno ju je odvući sa scene u imovinu igre i pretvoriti ju u predložak.

Pokretne prepreke se rade na sličan način kao i stacionarne. Jedino je potrebno dodati skriptu koja omogućuje kretanje same prepreke. Skripta FollowPath omogućuje da se prepreka kreće između dvije točke unutar scene.

```
public Transform Point1;
public Transform Point2;
private bool moveToPoint1;
public float Speed=1;
public float MaxDistanceToGoal = 0.1f;
private SpriteRenderer renderer;
```

Kod 12. Polja skripte FollowPath

Polja Point1 i Point2 su tipa Transform i odnose se na dvije točke u prostoru između kojih se prepreka kreće. Polje moveToPoint1 je tipa boolean te služi za provjeru treba li se prepreka kretati prema prvoj točki ili drugoj. Polje Speed se odnosi na brzinu kojom će se prepreka kretati unutar prostora, dok polje MaxDistanceToGoal se odnosi na udaljenost na kojoj prepreka mora biti od točke kako bi se krenula kretati prema drugoj točki. Polje renderer se odnosi na SpriteRenderer komponentu objekta i u ovoj skripti se koristi kako bi se okrenulo sliku prepreke kada se počne kretati prema drugoj točki.

```
public void Start()
{
    moveToPoint1 = true;
    renderer = GetComponent<SpriteRenderer>();
}
```

Kod 13. Start metoda skripte FollowPath

U Start metodi skripte FollowPath se vrijednost polja moveToPoint1 stavlja kao istinita kako bi se prepreka u početku kretala prema prvoj točki. Potom se u polje renderer dodaje referenca na SpriteRenderer komponentu same prepreke.

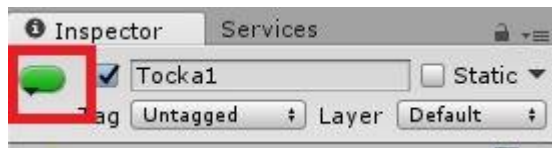
```
public void Update(){
    if (moveToPoint1)
    {
        transform.position = Vector3.MoveTowards(transform.position, Point1.transform.position, Time.deltaTime
* Speed);
        var distanceSquared = (transform.position - Point1.transform.position).sqrMagnitude;
        if (distanceSquared < MaxDistanceToGoal * MaxDistanceToGoal)
        {
            moveToPoint1 = !moveToPoint1;
            renderer.flipX = !renderer.flipX;
        }
    }
    else if (!moveToPoint1)
    {
        transform.position = Vector3.MoveTowards(transform.position, Point2.transform.position, Time.deltaTime
* Speed);
        var distanceSquared = (transform.position - Point2.transform.position).sqrMagnitude;
        if (distanceSquared < MaxDistanceToGoal * MaxDistanceToGoal)
        {
            moveToPoint1 = !moveToPoint1;
            renderer.flipX = !renderer.flipX;
        }
    }
}
```

Kod 14. Update metoda FollowPath skripte

Unutar Update metode, ponajprije se provjerava koja je vrijednost polja moveToPoint1, ukoliko je ona istinita prepreka se treba kretati prema prvoj točki. To čini tako da se pomoću

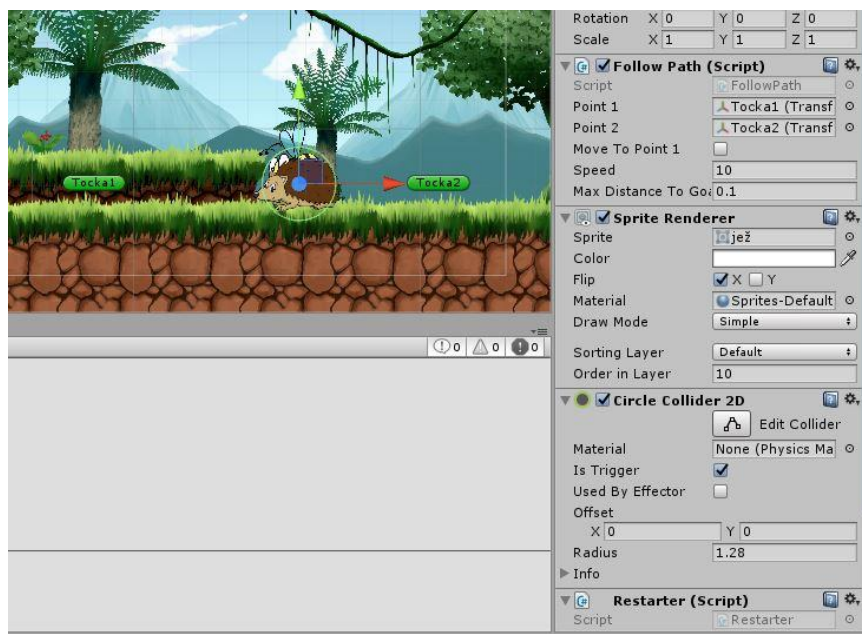
metode `MoveTowards` pomiče od svoje trenutne pozicije prema poziciji prve točke i to brzinom koja je zadana u polju `Speed`. Potom se preko udaljenosti između same prepreke i prve točke i polja `MaxDistanceToGoal` provjerava je li prepreka stigla do prve točke. Ukoliko je stigla, vrijednost polja `moveToPoint1` se mijenja u suprotnu, neistinitu vrijednost te se preko `SpriteRenderer` komponente slika prepreke horizontalno okreće za 180 stupnjeva. U slučaju da vrijednost polja `moveToPoint1` je neistinita, `Update` metoda radi sve isto samo usmjereno na drugu točku.

Za kreiranje predloška prepreke koja se kreće, potrebno je prvo napraviti objekt unutar kojeg će se nalaziti prepreka koja se kreće, te točke između kojih će se ta prepreka kretati. Za kreiranje praznog objekta, u glavnom izborniku se izabere `GameObject` i potom `Create Empty`. Nakon što je kreiran prazan objekt, postavi ga se na poziciju (0,0) te se njemu kreira dva dječja objekta. Dječje objekte se kreira tako da se u hijerarhiji desnim klikom stisne na objekt te se odabere `Create Empty`. Ti dječji objekti će predstavljati točke između kojih će se kretati prepreka. Točka se postavlja na određenu udaljenost od središnje točke roditeljskog objekta. Za uočavanje tih točaka u sceni, dobro je postaviti da imaju određenu ikonu. Za postavljanje ikone, potrebno je kliknuti na gumb koji je označen na sljedećoj slici.



Slika 23. Dodavanje ikone objektu

Nakon što su dodane točke između kojih će se prepreka kretati, potrebno je dodati dječji objekt koji će predstavljati prepreku. Za dodavanje određene prepreke, prvo se slika te prepreke iz imovine odvuče na roditeljski objekt u hijerarhiji te se postavi na poziciju (0,0). Potom mu se doda odgovarajući `Collider` kojem se vrijednost polja `Is Trigger` postavi kao istinita. Na kraju se doda skripta `FollowPath`. Unutar nje, potrebno je iz hijerarhije odvući točke između kojih će se prepreka kretati u odgovarajuća polja unutar inspektora te postaviti željenu brzinu kretanja prepreke. Na kraju, prepreku treba odvući u imovinu kako bi se napravio predložak.



Slika 24. Primjer pokretne prepreke

Uz prepreke, još jedna stvar koja je potrebna za kreiranje zabavnije igre je mogućnost prelaženja na sljedeću razinu. Za to će služiti skripta `LoadNextLevel` koja će učitati sljedeću scenu kada igrač uđe u Collider objekta koji ima tu skriptu na sebi.

```
public void OnTriggerEnter2D(Collider2D other)
{
    if (other.tag == "Player")
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
    }
}
```

Kod 15. OnTriggerEnter metoda skripte LoadNextLevel

Skripta pomoću `LoadScene` metode učitava sljedeću scenu, odnosno razinu u igri kada igrač uđe u Collider objekta koji sadrži tu skriptu. Za kreiranje predloška objekta koji učitava sljedeću razinu potrebna je slika koja će taj objekt predstavljati na sceni, Collider s vrijednosti polja `Is Trigger` kao istinitom te skripta `LoadNextLevel`. Potom je dovoljno taj objekt odvući u imovinu igre.

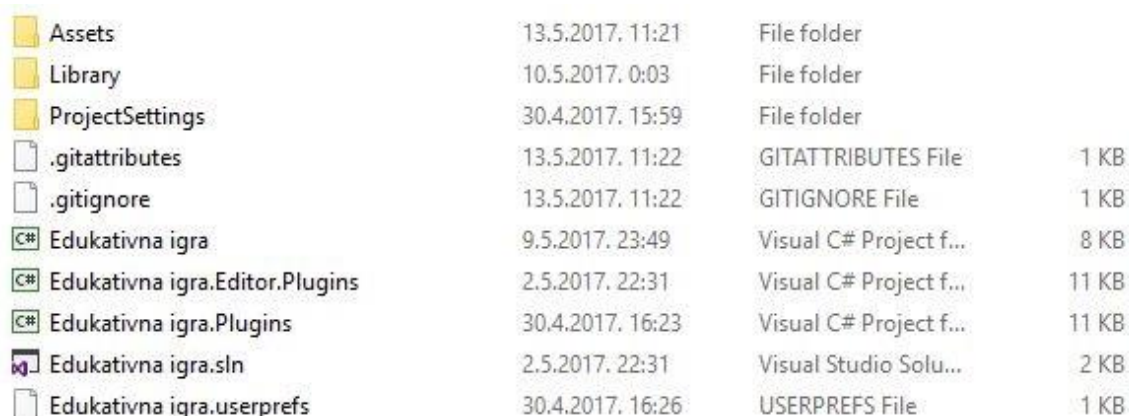
Ukratko, Unity 2D paket imovine pruža lika pomoću kojeg se može kretati unutar igre, dok imovina za izgradnju prostora omogućuje kreiranje prostora unutar kojeg se igrač kreće. Ploče s obavijestima i neutralni likovi omogućuju prikazivanje određenog sadržaja, dok prepreke i prelazak na sljedeću razinu omogućuju da igra bude zabavnija za igranje.

7. Kreiranje prezentacijske igre

Nakon što su pojašnjeni predlošci koji su potrebni za izgradnju prezentacijske igre, moguće je pojasniti kako se sama igra izgrađuje. U ovom poglavlju pokazat će se kako se izgrađuje prostor unutar kojeg se igrač može kretati. Potom će se prikazati kako se postavljaju ploče s obavijestima i neutralni likovi u prostor te kako im se dodaje sadržaj koji oni prikazuju. Na kraju će se pokazati kako se igri dodaju prepreke i prelasci na nove razine te kako se izgrađuje sama igra koju se može pokrenuti.

7.1. Početni projekt i kreiranje nove scene

Za kreiranje videoigre koju se može koristiti kao potporu u prezentiranju znanja, ponajprije je potrebno preuzeti početni projekt u kojem se nalaze potrebni predlošci za izradu igre. Početni projekt se može preuzeti sa stranice: <https://github.com/pavo76/Edukativna-igra>.

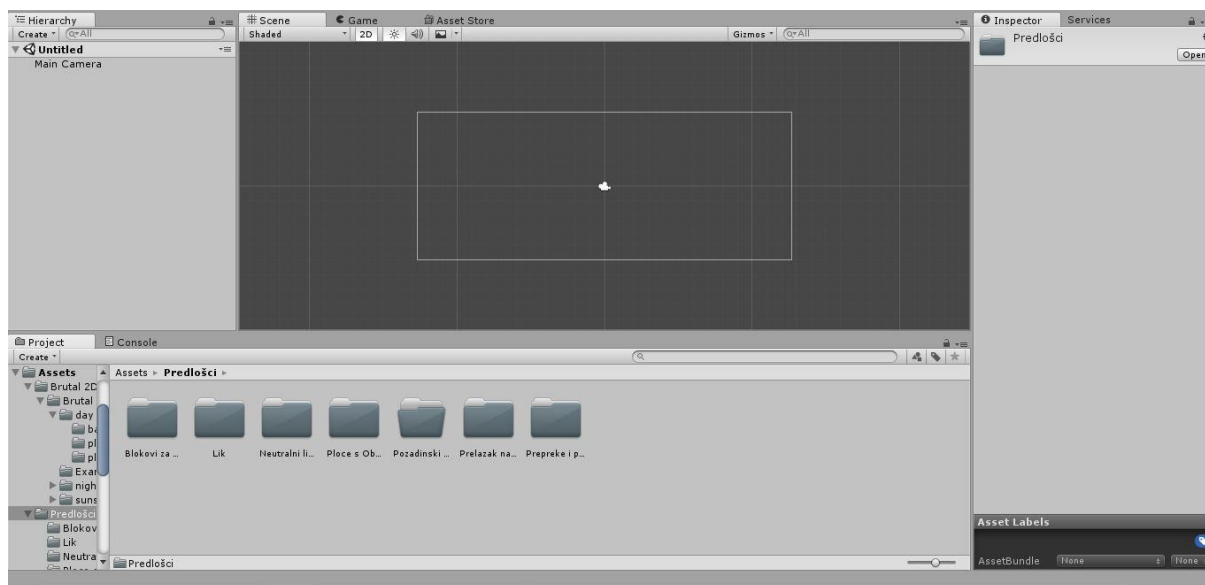


Assets	13.5.2017. 11:21	File folder	
Library	10.5.2017. 0:03	File folder	
ProjectSettings	30.4.2017. 15:59	File folder	
.gitattributes	13.5.2017. 11:22	GITATTRIBUTES File	1 KB
.gitignore	13.5.2017. 11:22	GITIGNORE File	1 KB
Edukativna igra	9.5.2017. 23:49	Visual C# Project f...	8 KB
Edukativna igra.Editor.Plugins	2.5.2017. 22:31	Visual C# Project f...	11 KB
Edukativna igra.Plugins	30.4.2017. 16:23	Visual C# Project f...	11 KB
Edukativna igra.sln	2.5.2017. 22:31	Visual Studio Solu...	2 KB
Edukativna igra.userprefs	30.4.2017. 16:26	USERPREFS File	1 KB

Slika 25. Mapa početnog projekta

Nakon što je projekt preuzet, unutar mape Assets može se otvoriti datoteka Start Scena kako bi se otvorio cijeli projekt u Unityju. Drugi način za otvaranje projekta je da se u početnom sučelju Unityja odabere Open, ili u glavnom izborniku Unityja pod File da se odabere Open Project. Potom je potrebno odabrati mapu u kojoj se nalazi početni projekt videoigre. Prilikom otvaranja projekta, otvara se scena na kojoj se posljednje radilo. Za kreiranje nove scene, potrebno je u glavnom izborniku pod File odabrati New Scene. Na taj način se otvara nova scena koja u sebi ima samo kameru. Unutar te prazne scene, potrebno je kreirati prostor igre u kojem će se igrač kretati, odati samog lika s kojim će se igrač kretati kroz igru te dodati ploče

s obavijestima i neutralne likove koji će prenositi određeni sadržaj. Po potrebi, u scenu je moguće dodati i prepreke da igra bude zabavnija, te objekt za prelazak na sljedeću razinu ukoliko igra ima više razina. U ovom primjeru pokazat će se kako se izgrađuje igra koja se može koristiti u prezentiranju o člankonošcima.

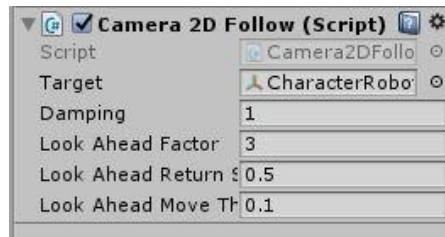


Slika 26. Nova scena

7.2. Kreiranje prostora i umetanje lika

Prilikom kreiranja scene, prvo što treba kreirati jest prostor same igre te umetnuti lika pomoću kojeg će se igrač kretati kroz prostor. Kako je za isprobavanje prostora potreban sam lik, prvo će se objasniti kako se dodaje lik na scenu i podešava kamera da ga prati. Drugim riječima, da bi se igrač mogao kretati kroz prostor, potrebno je prilikom izgradnje prostora isprobati može li se lik slobodno kretati kroz taj prostor.

Za dodavanje lika, potrebno je u imovini igre, unutar mape Predložci otvoriti mapu Lik te predložak s likom odvući na scenu. Samim time lik je spreman za kretanje kroz prostor igre. Međutim, potrebno je podesiti kameru da prati lika. Da bi kamera pratila lika, potrebno joj je dodati komponentu Camera 2D Follow. Unutar te komponente, pod poljem Target potrebno je iz hijerarhije odvući lika kako bi kamera pratila lika unutar igre. Na taj način je dodan lik na scenu te omogućeno da kamera prati lika.



Slika 27. Camera 2D Follow komponenta na kameri

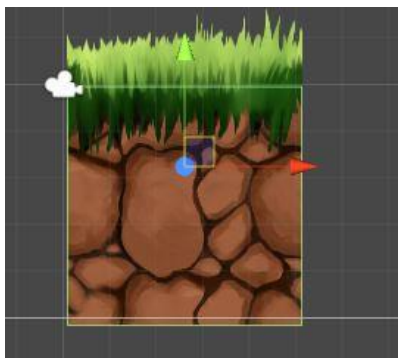
Nakon što je lik umetnut na scenu, može se preći na izgradnju samog prostora igre. Najbitniji dio kod izgradnje prostora igre je umetanje blokova po kojima će lik moći hodati. Predlošci s blokovima po kojima igrač može hodati se nalaze u mapi Blokovi za hodanje unutar mape. Predlošci unutar imovine igre. Kako su svi predlošci blokova podešeni na odgovarajuću poziciju u sceni, prilikom izgradnje prostora jedino je potrebno postavljati blokove jedan do drugoga i tako izgraditi prostor kroz koji se lik može kretati.

Za dodavanje određenog bloka na scenu, dovoljno je odvući ga iz imovine na scenu te ga postaviti na željenu poziciju. Za lagano pozicioniranje blokova unutar scene, najbolje je ući u način rada za pomicanje objekata tako da se pritisne na tipku **W** ili se u sučelju Unitya pritisne na gumb za način rada za pomicanje objekata koji se nalazi odmah ispod glavnog izbornika.



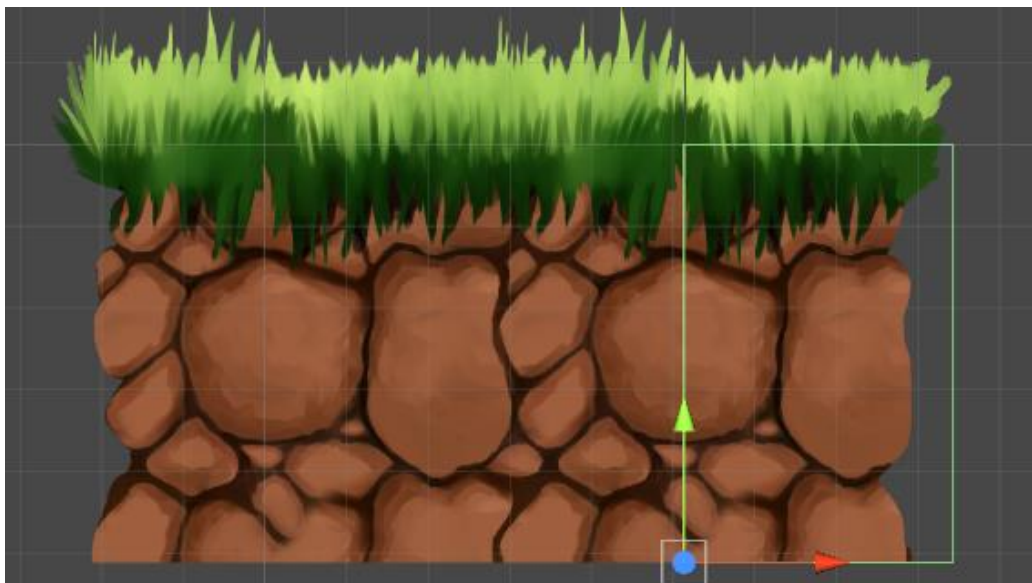
Slika 28. Gumb za način rada za pomicanje objekata

U tom načinu rada se prikazuju strelice za pomicanje po osima te kvadratić za slobodno pomicanje iznad odabranog objekta u sceni. Klikom i držanjem strelice moguće je pomicati objekt, odnosno blok po odgovarajućoj osi. Ukoliko se klikne i drži kvadratić, moguće je slobodno pomicati objekt unutar scene.



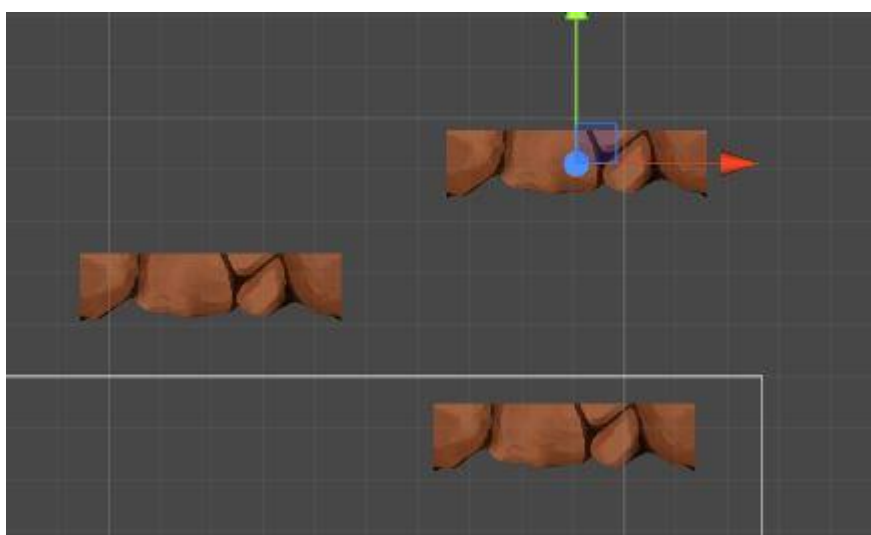
Slika 29. Strelice i kvadratić za pomicanje objekta

Za dodavanje još jednog bloka, jednako tako se može odvući predložak iz imovine na scenu. U slučaju da se želi postaviti isti predložak na scenu kao prethodni, dovoljno je da se odabere taj predložak na sceni te pritisnuti tipke **CTRL** i **D**. Na taj način se duplicira željeni blok, te ga se potom pozicionira tako da bude povezan s prethodnim blokom. Prilikom povezivanja blokova, potrebno je pripaziti da su vrhovi njihovih Collidera na jednakoj razini kako bi igrač nesmetano mogao proći s jednog bloka na drugi. Kako su u predlošcima Collideri blokova postavljeni tako da ako je dno dvaju objekata blokova na istoj razini, tada su i gornje stranice Collidera bloka na jednakoj razini. Drugim riječima, ukoliko je dno dvaju objekata blokova na istoj razini, igrač će nesmetano moći preći s jednog bloka na drugi. Za postavljanje dna objekata na jednaku razinu može se koristiti način rada za kopčanje vrhova (engl. Vertex snap) u koji se prelazi držanjem tipke **V**. U tom načinu rada, prvo se može odabrati jedan od vrhova objekta ili sama sredina tako da se pozicionira miša na željeni vrh. Pošto je ovdje potrebno postaviti dno objekta na istu razinu, odabire se donji lijevi vrh ako se blok povezuje s blokom koji mu je slijeva, ili donji desni vrh ukoliko ga se povezuje s blokom koji mu je s desna. Kada je miš pozicioniran na željeni vrh, klikne se na njega te se taj vrh odvuče do vrha drugog bloka s kojim želimo povezati prvi blok. Sam način rada kopčanja vrhova će spojiti vrhove dvaju blokova čim ih se približi jednog drugome te tako osigurati da su im dna na istoj razini, a time i osigurati da igrač može nesmetano prelaziti s jednog bloka na drugi.



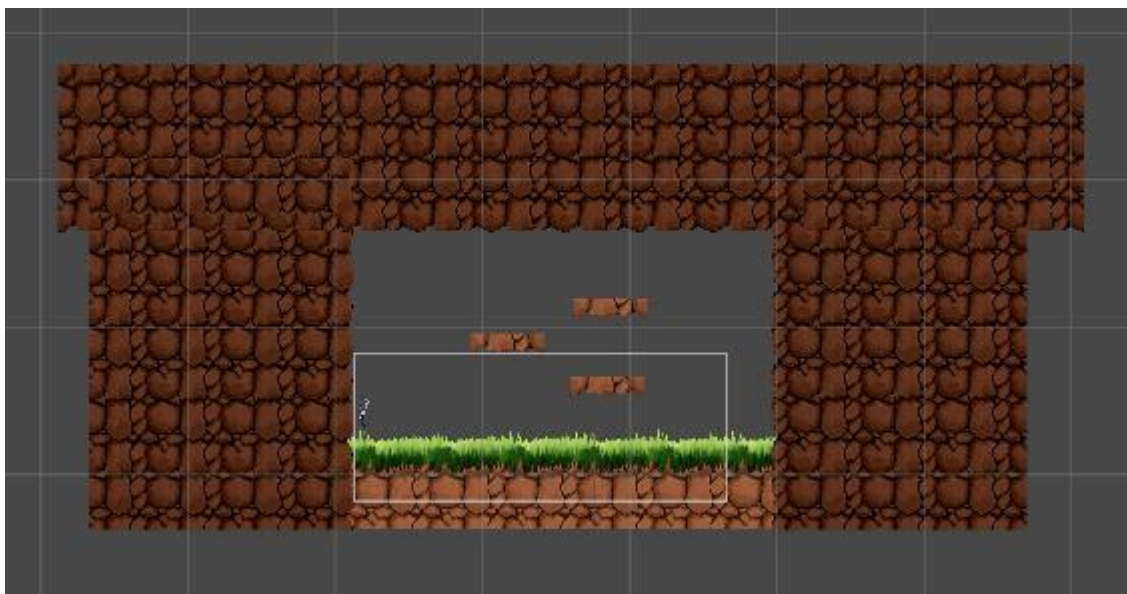
Slika 30. Kopčanje vrhova

Dodavanjem blokova na scenu te spajanjem njihovih vrhova moguće je izgraditi prostor za kretanje unutar igre. Za izgradnju tla se koriste blokovi čiji predlošci u nazivu imaju „srednji blok“ za blokove u sredini. Za blokove na rubu tla se koriste blokovi koji imaju u imenu „rubni blok desno“ i „rubni blok lijevo“ za odgovarajuće rubove. Za prostor u kojem se igrač može kretati prema gore ili dolje, dovoljno je posložiti manje odvojene blokove tako da je svaki viši ili niži od drugog te da igrač može skočiti s jednog na drugi kao što je prikazano na slici 32. Za te blokove se koriste predlošci koji imaju u imenu „mali blok“.



Slika 31. Postavljanje blokova za kretanje prema gore ili dolje

Posljednji dio koji treba dodati kod izgradnje prostora po kojem će se igrač kretati su blokovi koji će osigurati da igrač ne može „ispasti“ iz prostora. Za taj dio prostora se koriste predlošci koji u imenu imaju „granični blok“ te „lijevo“, „desno“ i „gore“ za odgovarajuće strane koje se treba postaviti granicu. Za donju granicu se koriste normalni blokovi po kojima igrač hoda.



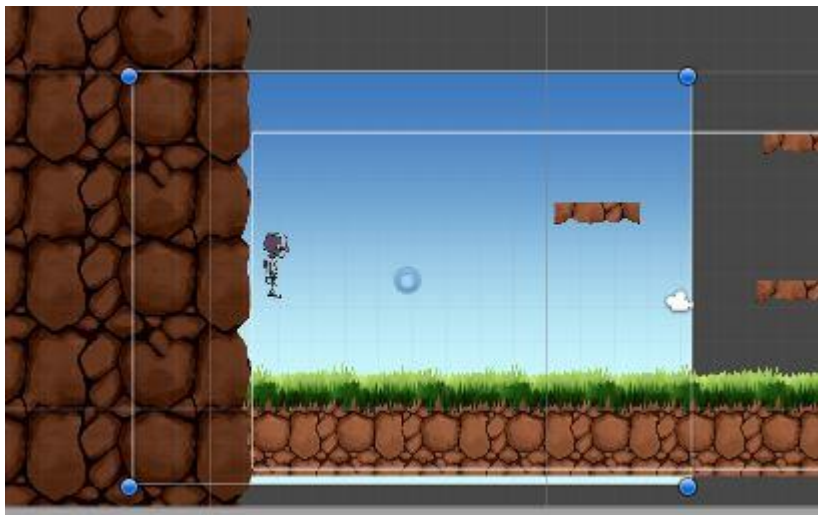
Slika 32. Prostor ograden graničnim blokovima

Nakon što su postavljeni blokovi za kretanje, mogu se dodati pozadinski i ukrasni blokovi kako bi igra djelovala zanimljivije te pojedini prostori razlikovali čime bi se omogućilo lakše povezivanje pojedinog sadržaja s određenim prostorom. Kako je poredak na sloju tih blokova već podešen u predlošcima, za postavljanje pozadinskih i ukrasnih blokova je dovoljno da ih se iz imovine odvuče na scenu te ih posloži po vlastitom ukusu. Predlošci za pozadinske i ukrasne blokove se nalaze u mapama Pozadinski blokovi i Ukrasni blokovi. Kod pozadinskih blokova je jedino nužno postaviti blok koji u imenu ima „krajnja pozadina“ kojeg se postavlja na scenu i potom uvećava tako da pokrije cijeli dio prostora za koji je on pozadina. To je potrebno napraviti kako se na kameri ne bi prikazivala praznina, već željena pozadina. Za uvećavanje objekta potrebno je prebaciti se u način rada za povećavanje objekata pritiskom na tipku **T** ili pritiskom na gumb za taj način rada kao što je prikazano na slici.



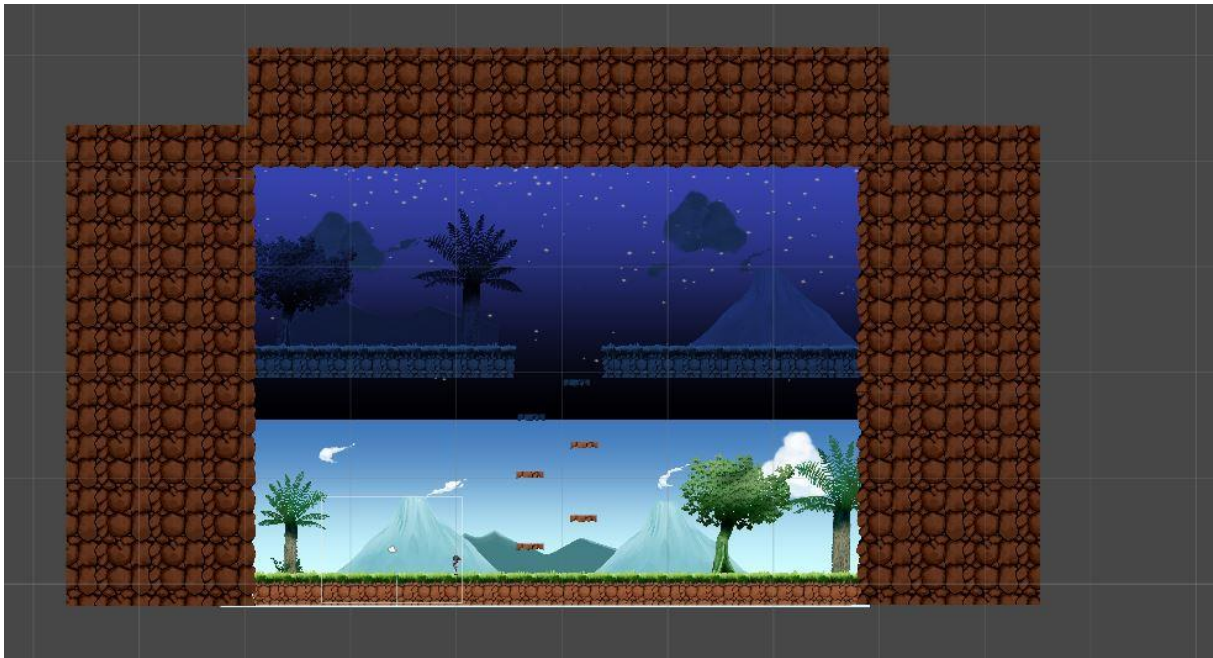
Slika 33. Način rada za povećavanje objekata

U tom načinu rada, dovoljno je da se pritisne na jedan od rubova objekta te se povlačenjem objekt uveća ili smanji na željenu veličinu. U slučaju krajnje pozadine, potrebno je objekt uvećati tako da pokrije cijeli prostor za koji je on pozadina.



Slika 34. Način rada za uvećavanje objekata

Ostatak pozadinskih i ukrasnih blokova se može postaviti po volji, no treba pripaziti da ih se ne stavi previše kako ne bi odvlačili pažnju od samog sadržaja. Budući da će se u primjeru raditi igra za prezentiranje sadržaja o člankonošcima, ta igra treba imati četiri dijela prostora: uvodni prostor, prostor s kukcima, prostor s paucima i prostor s klještarima. Igrač kreće iz uvodnog prostora te se iz tog razloga samog lika na sceni postavlja u uvodni prostor. Potom se iz uvodnog prostora treba moći preći u bilo koji od preostala tri prostora te time omogućiti dinamičnije prezentiranje pošto kod grananja nije određeno na koji će se sljedeći prostor preći, a uz to da bi se pokazala i veza između pojedinih sadržaja. U ovom slučaju da bi se prikazalo da su kukci i pauci međusobno povezaniji nego s klještarima, prostor s klještarima će se nalaziti desno od uvodnog prostora, a prostor s kukcima i prostor s paucima će se nalaziti iznad njih, svaki s jedne strane od mjesta grananja. Pored toga, za daljnji prikaz povezanosti, prostori s kukcima i paucima će imati blokove koji nose u imenu „noć“ a uvodni prostor i prostor s klještarima će imati blokove koji nose u imenu „dan“. Na taj način će se i samim vizualnim prikazom prostora omogućiti da se prikaže struktura između sadržaja. Navedeni prostor igre izgleda kao na slici 36.

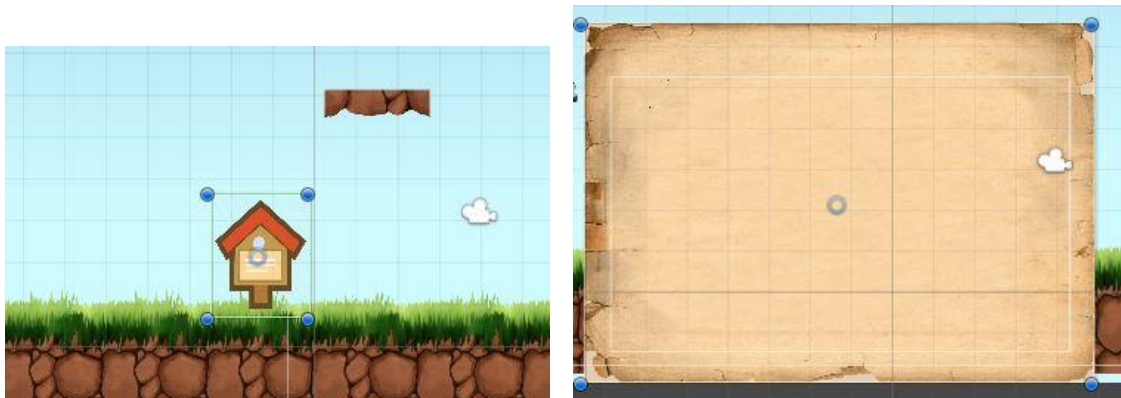


Slika 35. Krajnji prostor igre

7.3. Kreiranje ploča s obavijestima

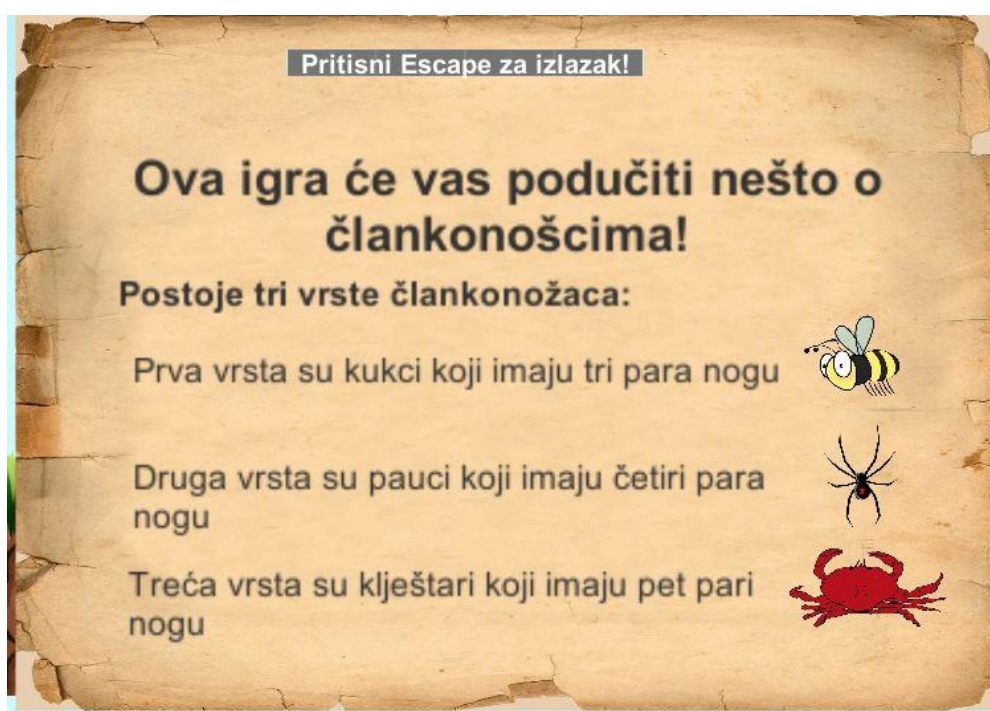
Nakon što je izgrađen prostor igre u kojem se igrač može kretati, potrebno je objasniti kako se umeće sam sadržaj koji se treba prikazati u igri. Taj sadržaj se prikazuje preko ploča s obavijestima i neutralni likovi. U ovom dijelu će se prikazati kako se kreiraju ploče s obavijestima te kako im se dodaje sadržaj koji one prikazuju.

Za dodavanje ploče s obavijestima u prostor potrebno je najprije predložak ploče odvući iz imovine na scenu na željenu poziciju. Nakon što je ploča postavljena u prostoru, za dodavanje sadržaja ploči, prilikom uređivanja se dječji objekt pozadine postavlja kao aktivan.



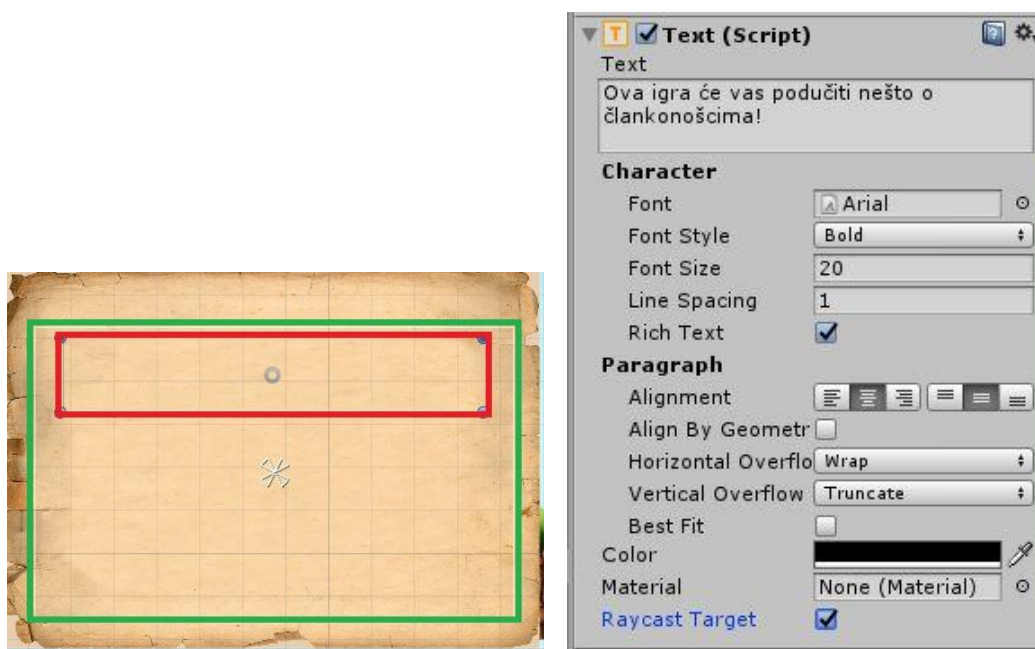
Slika 36. Ploča s obavijestima postavljena u prostoru(lijevo), ploča s aktivnom pozadinom (desno)

Nakon što je pozadina aktivirana, na ploču se mogu dodati dva tipa sadržaja, tekst i slike. Ponajprije će se opisati kako se dodaje tekst, a potom slike. Pošto je primjer igre o člankonošcima, za primjer će se pokazati kako se dodaje sadržaj na ploču koja će ukratko uvesti igrače u igru. Na vrhu ploče će pisati: „Ova igra će vas podučiti nešto o člankonošcima.“ Ispod toga će pisati: „Postoje tri vrste člankonožaca:“ Nakon toga će biti red ispod reda napisano, „Prva vrsta su kukci koji imaju tri para nogu“, „Druga vrsta su pauzi koji imaju četiri para nogu“ i „Treća vrsta su klješćari koji imaju pet pari nogu“. Pored tih tri rečenice će biti postavljena slika predstavnika pojedine vrste. Dakle primjer ploče će imati pet tekstualnih objekata te tri slike.



Slika 37. Primjer ploče s obavijestima

Za dodavanje teksta na ploču, potrebno je desnim klikom kliknuti na dječji objekt Canvas te pod UI odabrati Text. Za razliku od dodavanja Texta na samu ploču, kada se cijeli Canvas postavio kao zaseban objekt, u ovom slučaju će se samo dodati objekt Text kao dječji objekt Canvasu. Taj objekt predstavlja tekstni okvir unutar kojeg će se ispisati određeni tekst. Za podešavanje veličine tekstnog okvira, dovoljno je prebaciti se u način rada za povećavanje objekata pritiskom na tipku **T**, a za podešavanje pozicije se prebaci u način rada za pomicanje objekata pritiskom na tipku **W**. Nakon što je podešen tekstni okvir, za unosenje i podešavanje teksta je potrebno podesiti vrijednosti određenih polja unutar Text komponente u inspektoru.



Slika 38. Podešavanje tekstnog okvira unutar Canvasa (lijevo), Text komponenta unutar inspektora (desno)

Polje Text se odnosi na sam tekst koji želimo prikazati. U ovom slučaju je napisan tekst koji odgovara naslovu primjera. Umjesto teksta koji je napisan na slici, za ostale tekstove primjera se samo upiše odgovarajući tekst.

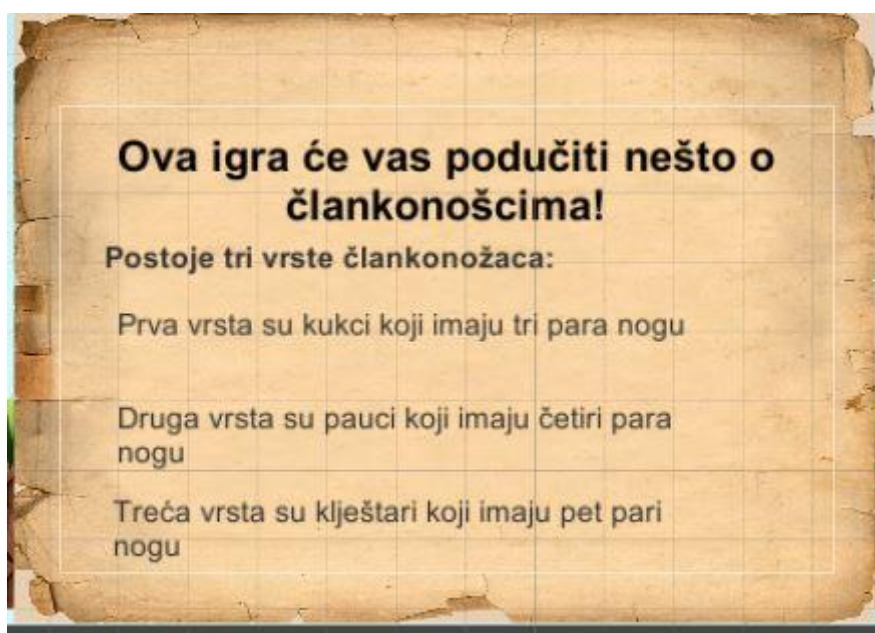
Unutar skupine polja Character, polje Font se odnosi na font teksta koji će se prikazivati, Font Style na stil teksta, Font size na njegovu veličinu dok Line Spacing se odnosi na razmak između redaka. Polje Rich Text se odnosi na omogućavanje detaljnijeg uređenja teksta tako da se određeni dijelovi stave između oznaka. Ako želimo da određeni dio teksta bude podebljan, stavi ga se između oznaka `` i ``. S druge strane, ako tekst treba biti ukošen, stavi ga se između oznaka `<i>` i `</i>`.¹

U skupini polja Paragraph, polje Alignment omogućuje podešavanje kako će se tekst poravnati unutar tekstnog polja. Polje Horizontal Overflow određuje hoće li tekst preći izvan tekstnog okvira ukoliko je vrijednost postavljena na Overflow ili će ga se zadržati unutar granica tekstnog okvira ako ga se postavi na Wrap. Jednako tako, polje Vertical Overflow će odrezati višak teksta ako se njegova vrijednost postavi na Truncate, ili prikazati sav tekst ako ga se postavi na Overflow. Za najlakši unos teksta koji ne odlazi previše u širinu, ali prikazuje se u potpunosti Horizontal Overflow se postavlja na Wrap a Vertical Overflow na Overflow.

¹ Više o oznakama na: <https://docs.unity3d.com/Manual/StyledText.html>

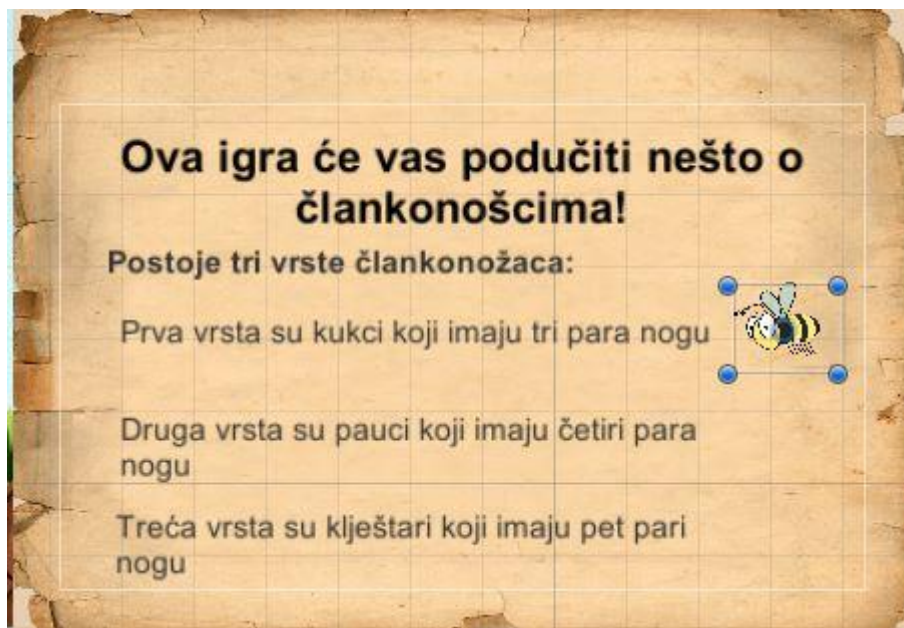
Uz to, moguće je postaviti polje Best Fit kao istinito te na taj način podesiti tekst da ispuni cijeli tekstni okvir što je preciznije moguće. Na kraju, unutar polja Color se određuje boja samog teksta.

Ponavljajući prethodno navedeni postupak može se dodavati tekst na ploču. Iako bi se mogao sav tekst dodati preko jednog objekta, radi lakšeg uređenja pojedinih tekstova i lakšeg organiziranja sadržaja na ploči, bolja je praksa da se svaka cjelina teksta doda kao zaseban objekt. Drugim riječima, u slučaju primjera unutar objekta Canvas će se dodati 5 dječjih objekata koji su tipa Text. Pored toga, unutar hijerarhije je moguće pojedine objekte teksta preimenovati kako bi se lakše moglo prepoznati koji se objekt odnosi na koji tekst.



Slika 39. Ploča s dodanim tekstom

Za dodavanje slike na ploču s obavijestima, postupak je nešto jednostavniji. Prvo je potrebno željenu sliku dodati u imovinu igre tako da ju se odvuče u sučelje s imovinom. Potom je potrebno sliku iz imovine odvući na objekt pozadine kako bi ta slika bila dječji objekt pozadine. Slici je potom potrebno postaviti da je poredak na sloju 100 ili više kako bi se prikazala ispred pozadine sadržaja ploče kojoj je poredak na sloju 90. Potom je potrebno podesiti veličinu slike u način za podešavanje veličine objekta te ju postaviti na željeno mjesto na ploči s obavijestima. Kod primjera sve slike se postavljaju pored teksta, no sliku se može postaviti bilo gdje dokle god je ona na pozadini ploče kako bi se osiguralo da se slika prikaže unutar igre.



Slika 40. Ploča s dodanom slikom

Ponavljajući ovaj postupak može se dodati neodređeni broj slika, no treba pripaziti da ih ne bude previše kako ne bi ustvari odvlačile pažnju učenika. Isto tako i kod teksta treba pripaziti da ga ne bude previše kako učenicima ne bi ubrzo postalo dosadno i da prestanu pratiti prezentaciju. Drugim riječima, ploče služe za prikaz općenitog sadržaja kojeg nema previše na jednoj ploči. Jednako tako, nizanjem ploča s obavijestima u igri može se postići privid kao da igra ima prikaz slajdova gdje svaka ploča predstavlja jedan slajd. Nakon što su dodani tekst i slike na ploču, potrebno je ponovno postaviti objekt pozadine kao neaktivan kako ne bi smetala prilikom daljnjeg kreiranja igre te da se pravilno sadržaj prikaže kada igrač dođe do ploče.

Uz prikaz sadržaja koji treba prezentirati, ploče s obavijestima se mogu koristiti za usmjeravanje igrača unutar igre, odnosno kao putokaz. Na raskrižjima unutar igre, mogu se postaviti ploče s obavijestima koje govore igraču što mogu pronaći u kojem smjeru. U primjeru bi se tako na raskrižju postavila ploča koja govori igraču da nastavi desno ako želi saznati više o klješćarima, a da krene prema gore ako ga zanimaju pauzi i kukci. Nadalje, na gornjoj razini još jedna ploča može uputiti igrača da krene lijevo kako bi saznao nešto o kukcima, a desno da sazna više o paucima. Na taj način se omogućuje da prezentacija ima dinamičan tok, te da ovisno o potrebi prezentacije se nastavi u smjeru onog sadržaja koji se želi dalje prezentirati.

Ukratko, kod ploča s obavijestima se može postaviti određeni sadržaj u tekstualnom ili slikovnom obliku te na taj način predstaviti sadržaj koji je potrebno prezentirati. U slučaju ploča se ne smije previše teksta stavljati kako se ne bi zamaralo učenike, niti se smije staviti previše slika kako se ne bi odvrćala pozornost učenika prilikom prezentiranja. U slučaju kada treba prikazati više teksta, odnosno nešto detaljnije pojasniti koriste se neutralni likovi koji putem dijaloga predstavljaju određen sadržaj.

7.4. Kreiranje neutralnih likova

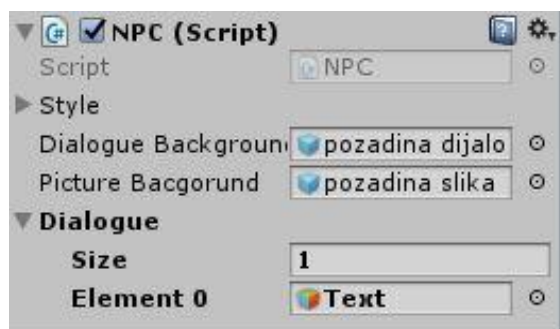
Postupak kreiranja neutralnih likova je sličan kao i kod ploča s obavijestima. Razlika je u tome što se pojedini dio dijaloga kreira zasebno te je potrebno odrediti kojim redom se dijalog prikazuje. Uz to, razlika se javlja kod prikazivanja slika uz dijalog.

Dodavanje dijela dijaloga se radi istim postupkom kao i kod dodavanja teksta na ploču s obavijestima. Prije dodavanja teksta, potrebno je pozadinu dijaloga postaviti kao aktivnu da se može podesiti tekst na njoj. Za dodavanje teksta potrebno je desnim klikom kliknuti na Canvas dječji objekt unutar objekta pozadina dijaloga. Potom je potrebno podesiti tekstni okvir unutar Canvasa, a pošto dijelovi dijaloga imaju samo jedan tekst, može se podesiti da tekstni okvir zauzima većinu Canvasa. Nakon što je podešena veličina i pozicija tekstnog okvira, potrebno je upisati sam tekst dijaloga, urediti font tog teksta te ga poravnati unutar okvira. Nakon što je unesen tekst dijaloga, samom objektu tog teksta je potrebno dodati i pomoćnu skriptu HasPictureCheck. Unutar te komponente u inspektoru, polje HasPicture se postavlja kao istinito ukoliko je tom dijelu dijaloga priložena i slika. U protivnom, to polje se ostavlja kao neistinito.



Slika 41. Dio dijaloga bez slike

Na kraju, u hijerarhiji se ponajprije označi objekt neutralnog lika te se iz dječjih objekata odvuče objekt teksta u polje Dialogue s listom dijaloga unutar komponente NPC. Na kraju se taj objekt postavi kao neaktivan te se istim postupkom kreira sljedeći dio dijaloga.



Slika 42. Komponenta NPC s dodanim dijelom dijaloga

Pošto su dijelovi dijaloga većinskim dijelom isti, da se skрати postupak dodavanja novog dijela dijaloga, dovoljno je da se odabere prethodni dio dijaloga te pritiskom na tipke **CTRL** i **D** se duplicira prethodni tekst te mu se promijeni sadržaj teksta. Potom se novi dio dijaloga doda u polje Dialogue kao sljedeći dio koji se treba pokazati.

Kad se uz dio dijaloga želi prikazati i slika, postupak dodavanja teksta je isti. Prva razlika je što se u pomoćnoj skripti HasPictureCheck vrijednost polja HasPicture postavi kao istinita pošto će taj dio dijaloga imati priloženu sliku. Sljedeće što treba jest postaviti objekt pozadine slike kao aktivan kako bi se mogao podesiti prikaz slike. Nakon što je pozadina slike aktivna, objektu teksta dijaloga se kao dječji objekt dodaje željena slika. Ponajprije se željenu sliku dodaje u imovinu igre, a potom se slika odvuče na objekt teksta kako bi se ona postavila kao

dječji objekt te se njen poredak na sloju postavlja kao 100 ili više. Potom se podesi pozicija slike u odnosu na pozadinu slike te se podesi većina slike po želji. Iako se ovim postupkom može dodati i više slika, kako bi one čim manje odvlačile pažnju dobro je ne staviti ih puno.



Slika 43. Dio dijaloga sa slikom

Na kraju se objekt teksta, koji ovog puta u sebi ima i sliku kao dječji objekt, odvuče u polje Dialogue kako bi ga se postavilo kao sljedeći dio dijaloga te se njega i pozadinu slike stavi kao neaktivne kako bi se mogao dalje uređivati dijalog.

Ponavljajući prvi postupak kada se dodaje samo tekst, i drugi postupak kada se dodaje i slika može se kreirati dijalog s neodređenim brojem dijelova. Nakon što su napravljeni svi potrebni dijelovi dijaloga, sve objekte dijelova te pozadine dijaloga i slike se postavi kao neaktivne te na taj način završi kreiranje neutralnog lika s dijalogom. Iako se na ovaj način može prikazati iznimno velika količina teksta razdvojena na dijelove, dobro je pripaziti da količina sveukupnog dijaloga ne bude prevelika kako ne bi čitanje postalo naporno učenicima i time ih demotiviralo za praćenje prezentacije.

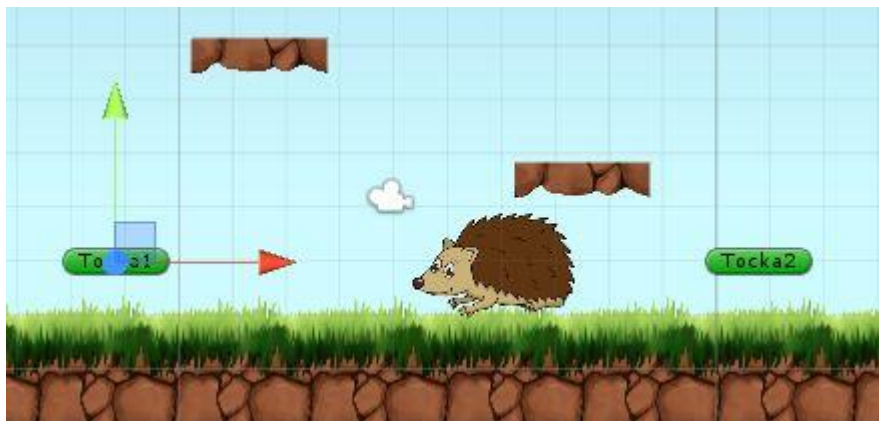
Kod prezentiranja sadržaja putem neutralnih likova, omogućeno je da učenici sadržaj povežu i sa samim likom koji ga prenosi. Time se uz postavljanje lika u određenu okolinu u prostoru olakšava pamćenje sadržaja.

Ukratko, za dodavanje dijelova dijaloga kod neutralnih likova potrebno je prvo postaviti predložak neutralnog lika u prostor gdje igrač treba naići na njega. Potom se kao dječji objekti dodaju dijelovi dijaloga koji su tima Text te se uređuje njihov sadržaj u odnosu na pozadinu dijaloga. Ako dio dijaloga uz sebe ima i sliku, slika se postavlja kao dječji objekt unutar objekta samog teksta te se unutar pomoćne skripte HasPictureCheck postavlja vrijednost polja HasPicture kao istinita. Uz to, sama slika se podešava u odnosu na pozadinu slike. Na kraju se

dijelovi dijaloga redom dodaju u polje Dialogue koje je lista dijelova dijaloga kako bi se oni mogli ispravnim redom prikazati. Ako je pak potrebno promijeniti redoslijed prikazivanja dijaloga, to se postiže povlačenjem dijelova unutar liste na odgovarajuću poziciju. U slučaju brisanja dijela dijaloga, potrebno je objekt djela obrisati iz hijerarhije te ponovno urediti poredak u listi.

7.5. Dodavanje prepreka te izrada igre

U slučaju da treba kreirati igru u kojoj se nalaze i prepreke te ima više scena potrebno je dodati i prepreke unutar scene te objekt za prelazak na sljedeću razinu. Za dodavanje prepreka dovoljno je predložak prepreke odvući na scenu iz mape Prepreke i protivnici te postaviti na željenu poziciju u prostoru. Pored toga, može se postaviti da postoje rupe u tlu te se ispod tla postavi prepreka dno koju je potrebno proširiti tako da se proteže duž cijelog dna. U tom slučaju bi igrač umro upadne li u rupu te bi iznova krenuo prelaziti trenutnu razinu. Kod prepreka koje se kreću, potrebno je postaviti i krajnje točke njihova kretanja. To se postiže tako da se unutar scene u način za pomicanje objekata postavi dječje objekte krajnjih točaka na željene pozicije.



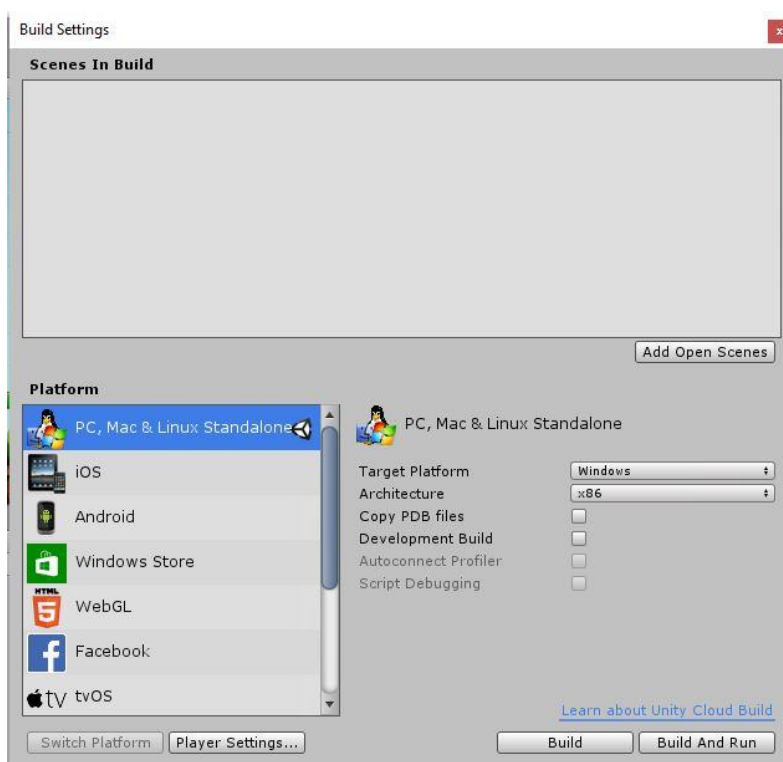
Slika 44. Pozicioniranje krajnjih točaka objekta

U slučaju izrade igre za prezentiranje znanja, dobro je ne postavljati prepreke jer se na taj način može produljiti vrijeme prelaska s jednog sadržaja na drugi, a samim time i izgubiti vrijeme potrebno za prezentiranje cjelokupnog sadržaja igre.

Ako se radi igra s više razina, svaka scena predstavlja jednu razinu te je na kraj prethodne scene potrebno postaviti objekt za prelazak na sljedeću razinu. Za to je dovoljno da se iz

mape Predlošci u imovini igre na scenu odvuče predložak pod imenom „Kraj“ te ga se postavi na željenu poziciju.

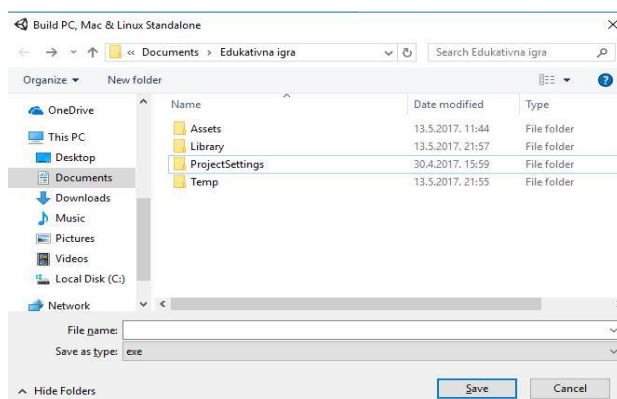
Nakon što je scena izgrađena, scenu se spremi pritiskom na tipke **CTRL** i **S** ili pritiskom na gumb Save Scene u glavnom izborniku pod File. Radi bolje preglednosti, scena se sprema u odvojenu mapu pod imenom Scene unutar imovine igre. Nakon što su sve scene igre spremljene, potrebno je izgraditi samu igru koju se može pokrenuti. Za izradu igre, potrebno je u glavnom izborniku Unityja, pod File pritisnuti na Build Settings. Potom se otvara sučelje za izradu igre.



Slika 45. Sučelje za izradu igre

Prvo što treba pri izradi igre napraviti je postaviti scene igre u Scenes In Build prozor sučelja. Najjednostavniji način dodavanja scene je da se iz imovine iz mape Scene odvuče scenu u prozor. Ukoliko igra ima više razina, postavljaju se sve scene u prozor željenim redoslijedom, gdje je najgornja scena prva koja će se pokrenuti, a najdonja scena zadnja. Potom je potrebno odabrati platformu za koju će se izgraditi igra. U ovom slučaju željena platforma je osobno računalo tako da se odabere PC, Mac & Linux Standalone. Ostale postavke se može ostaviti kako su postavljene te se pritisne na gumb Build. Potom se otvara sučelje za odabir mape gdje

će se postaviti igra te sam naziv igre koju se može pokrenuti. Na kraju se klikne na gumb Save te se pokrene izrada igre nakon koje je igra spremna za pokretanje.



Slika 46. Sučelje za odabir mape i imenovanje igre

NAME	DATE MODIFIED	TYPE	SIZE
primjer_Data	13.5.2017. 22:10	File folder	
primjer	30.3.2017. 9:20	Application	17.712 KB

Slika 47. Mapa napravljene igre

Nakon što je igra napravljena, u njejoj mapi se nalazi mapa s podacima igre te datoteka za pokretanje igre.

8. Korištenje videoigre u nastavi i moguća proširenja

Nakon što je objašnjeno kako se izrađuje obrazovna igra, potrebno je pojasniti na koji način se može ona koristiti. U ovom poglavlju će se prikazati u kojim dijelovima nastave se mogu koristiti videoigre napravljene prema ovom radu. Potom će se usporediti postupak izrade te korištenje videoigara s izradom i korištenjem PowerPointa. Na kraju će se iznijeti koja su moguća proširenja za prethodno predstavljen način izrade videoigara.

8.1. Načini korištenja videoigre u nastavi

Iako su se prije videoigre u obrazovanju koristile u svrhu vježbanja, videoigre predstavljene u ovom radu se ne mogu koristiti u svrhu vježbanja. Primarna svrha videoigre čija je izrada predstavljena u radu je da se koristi prilikom prezentiranja znanja. Drugim riječima, videoigra se može koristiti u nastavnoj etapi obrade novih nastavnih sadržaja. Kada se igra izrađuje u svrhu potpore pri prezentiranju znanja, potrebno je da se čim lakše prelazi s jednog sadržaja na drugi. Iz tog razloga, kod izrade je potrebno napraviti da je prostor između pojedinih sadržaja čim manji kako bi vrijeme prelaska s jednog sadržaja na drugi bilo čim kraće. Uz to, kod takve igre je dobro izbjegavati umetanje prepreka jer se time može nepotrebno odužiti vrijeme potrebno da se dođe do sljedećeg sadržaja koji treba prezentirati. Međutim, treba napomenuti da ovaj način izrade videoigara nije pogodan za sve vrste gradiva kao što niti samo e-učenje nije pogodno. Tako su društvene i humanističke znanosti pogodnije za korištenje ovakvog oblika prezentiranja sadržaja naspram prirodoslovnih i tehničkih znanosti.

Pored korištenja videoigre u obradi novih nastavnih sadržaja, videoigre se mogu izraditi i za korištenje u nastavnoj etapi ponavljanja. U ovom slučaju, videoigra se može izraditi da ima više razina te unutar njih se mogu postaviti prepreke. Potom se u igru ubacuju određeni sadržaji koje učenici trebaju ponavljati i to s razmakom između svakog sadržaja. Na taj način, učenici mogu igrati igru te u određenim razmacima unutar igre ponavljati sadržaj. Ovdje ne treba izbjegavati umetanje prepreka ili držati prostor između sadržaja jer na taj način se osigurava da prilikom ponavljanja učenici ne budu prepravljani sadržajem i da ne izgube motivaciju za ponavljanje. Pored toga, igra izrađena na ovaj način se može koristiti i u nastavnoj etapi motivacije tako da se kroz igru predstavi određeni sadržaj koji će se učiti na satu ili ponovi sadržaj s prethodnog sata.

Ukratko, videoigre koje se izrađuju na način predstavljen u radu se mogu koristiti u etapama obrade novih nastavnih sadržaja i ponavljanja. U obradi novih nastavnih sadržaja se koriste kao potpora prilikom prezentiranja znanja, a kod ponavljanja se koriste da se između ponavljanja sadržaja učenici igraju i tako ostanu motivirani za ponavljanje.

8.2. Usporedba s PowerPointom

Iako su se određene razlike između korištenja PowerPoint prezentacija i videoigara kod prezentiranja prethodno prikazale, ovdje će se iznijeti kratka usporedba kako bi se razlike mogle vidjeti na jednom mjestu.

Prva od razlika je vrijeme pripreme sadržaja. Već u samom e-učenju jedan od nedostataka je povećano vrijeme pripreme za učitelja naspram tradicionalnih oblika učenja. Kod PowerPointa, vrijeme pripreme nije znatno duže zbog jednostavnosti programa te mogućnosti lakog recikliranja sadržaja. S druge strane, kod izrade videoigara vrijeme pripreme je znatno veće naspram PowerPoint prezentacija i još više u usporedbi s tradicionalnim oblicima učenja. Pored toga, u izradi videoigre su mogućnosti recikliranja manje nego kod PowerPointa tako da su mogućnosti uštede vremena manje. Dakle, izrada videoigara zahtjeva više vremena pripreme naspram PowerPoint prezentacija.

S druge strane, e-učenje podiže motiviranost učenika i privlači pozornost različitim sadržajima. Iako se u PowerPoint prezentacije lagano može ubaciti razne multimedijske sadržaje, njihovim pretjeranim korištenjem može doći do odvlačenja pažnje. Uz to, lako se može dogoditi da prezentacija bude monotona i time učenici izgube motivaciju za učenje. Kod videoigara se lakše može izbjeći monotonost te time izbjeći gubljenje motivacije učenika. Međutim, i kod videoigara se pretjeranim ubacivanjem raznih sadržaja može odvući pažnja učenika od bitnog sadržaja. Drugim riječima, kod videoigara se može lakše postići motiviranost učenika i izbjeći monotonost zbog koje bi učenici izgubili tu motiviranost.

Kod prikaza strukture sadržaja, PowerPoint je ograničen svojom linearnom naravi te samim time je i ograničena mogućnost prikaza strukture sadržaja i povezanosti između njegovih pojedinih dijelova. Kako se u videoigri sadržaj nalazi unutar prostora te određene vizualne okoline, moguće je na temelju pozicije u prostoru te okoline u kojoj se sadržaj nalazi prikazati strukturu i povezanost sadržaja.

Prilikom prezentiranja, koristeći se PowerPoint prezentacijama, prelazak s jednog slajda na drugi je kratak te je time i prelazak s jednog sadržaja na drugi znatno kraći. Uz to, za prelazak je dovoljno pritisnuti samo jednu tipku što učitelju daje više slobode za kretanje tokom prezentiranja. S druge strane, u videoigrama je prelazak od jednog sadržaja do drugog nešto veći jer se učitelj kao igrač mora kretati kroz prostor da bi došao do određenog sadržaja. Uz to, pošto učitelj mora upravljati likom, njegova sloboda kretanja po učionici je ograničena. Iako se prostor između sadržaja može umanjiti, prelazak s jednog sadržaja na drugi će uvijek biti znatno brži i lakši kod PowerPoint prezentacija.

S druge strane, kako su PowerPoint prezentacije linearne naravi, znatno je manja mogućnost dinamičnog prezentiranja i više sama prezentacija uvjetuje koji će se sadržaj sljedeći prezentirati. Kako se u videoigrama sadržaj nalazi unutar prostora, moguće je napraviti odabir u koji prostor će se sljedeće krenuti i time odabrati koji će se sadržaj sljedeće prezentirati ovisno o situaciji u nastavi.

Na kraju, kod korištenja PowerPointa postoji mogućnost da učitelj više bude okrenut prema samoj prezentaciji i time izgubi pozornost učenika te on sam ne obraća pažnju na povratne informacije koje mu učenici šalju. Kod korištenja videoigara je ta mogućnost jednako velika prilikom prenošenja sadržaja, a nešto veća prilikom prelaska s jednog sadržaja na drugi pošto učitelj mora upravljati likom u igri.

Ukratko, za korištenje videoigara je potrebno više vremena za pripremu te je kod prezentiranja sadržaja veći prelazak s jednog sadržaja na drugi čime se može izgubiti vrijeme potrebno za prezentiranje. S druge strane, videoigre omogućuju da učenici budu više motivirani tokom prezentacije te pružaju mogućnost da prezentacija bude dinamičnija te da se prezentira sadržaj ovisno o prilici u nastavi.

8.3. Moguća proširenja

Iako se na način predstavljen u radu može prezentirati bilo koji sadržaj, postoji nekoliko proširenja koja se mogu napraviti na videoigri kako bi njena uporaba bila djelotvornija ili da se može koristiti u druge svrhe.

Prvo od proširenja je umetanje što više vizualno različitih predložaka za izgradnju prostora, predložaka za ploče s obavijestima te predložaka za neutralne likove. Na taj način se može

postići veća povezanost sadržaja s njegovom okolinom te time omogućiti učenicima lakše pamćenje tog sadržaja.

Sljedeće moguće proširenje je izrada drugih žanrova videoigara osim 2D platformerske igre. Različiti žanrovi videoigara omogućavaju manju monotonost, dok izrada videoigara u tri dimenzije omogućuje bolji prikaz strukture sadržaja.

Posljednje moguće proširenje na igru jest dodavanje programske logike i predložaka za provjeru znanja što bi omogućilo da se ovakve videoigre ne koriste samo u etapama motivacije, obrade novih nastavnih sadržaja i ponavljanja, već i u etapama vježbanja i provjere znanja.

Ukratko, dodavanjem raznih vizualnih sadržaja se može omogućiti da učenici lakše upamte sadržaj koji se uči, dok dodavanjem raznih žanrova u igru ili proširivanje prostora igre na tri dimenzije se može dalje umanjiti monotonost i bolje prikazati struktura sadržaja koji se uči. Uz to, dodavanjem logike za provjeru znanja se može omogućiti da se videoigre koriste u bilo kojoj etapi nastavnog sata.

9. Zaključak

Danas se u obrazovanju prilikom prezentiranja znanja uglavnom koristi Microsoftov PowerPoint za izradu prezentacija koje se potom koriste kao potpora u prezentiranju znanja. Postoji nekoliko alternativnih rješenja za prezentiranje znanja, no one su slične PowerPointu. Jedan od alata koji se također koristi u obrazovanju su videoigre, no one se ne koriste prilikom prezentiranja znanja. Do sada su se videoigre u obrazovanju primarno koristile u svrhu vježbanja preko igara vježbe i prakse ili u svrhu omogućavanja učenicima da pobliže iskuse sadržaj koji se uči preko simulacija. Cilj ovog rada je bio predstaviti način izrade videoigara koje se mogu koristiti kao potpora prilikom prezentiranja znanja. Kako je česta prepreka kod izrade videoigara potrebno znanje o programiranju u ovom radu se predstavio način na koji se mogu izraditi videoigre bez znanja programiranja. Za postizanje tog cilja se koristio Unity igrin pokretač koji je besplatan. Uz to, Unityjeva tržnica imovinom te koncept predložaka omogućuju da se prilikom izrade videoigre koriste već izrađeni objekti koji u sebi nose potrebnu programsku logiku. Kod izgrađivanja videoigre koja se koristi prilikom prezentiranja znanja su potrebni lik kojim se igrač može kretati unutar igre, prostor unutar kojeg će se kretati, objekti koji će prikazivati određen sadržaj te prepreke kako bi se igru učinilo zabavnijom. Kao objekte za prikazivanje sadržaja su se koristile ploče s obavijestima koje prikazuju određen sadržaj kao što u PowerPoint prezentaciji to prikazuje slajd te neutralni likovi koji određen sadržaj prikazuju kroz dijalog. U poglavlju 6. Podloga za kreiranje prezentacijske igre se prikazalo kako se mogu izraditi predlošci za prethodno navedene objekte, dok u 7. poglavlju Kreiranje prezentacijske igre se prikazao postupak izrade igre od postojećih predložaka. Na temelju izrade videoigre prema uputama iz 7. poglavlja se može uočiti kako izrada takvih videoigara zahtjeva više vremena pripreme te može potrošiti dio vremena potrebnog za prezentiranje na prelaženje s jednog sadržaja na drugi. Međutim, pomoću videoigre se može jasnije prikazati struktura određenog sadržaja te postići veća motiviranost kod učenika. Uz to, videoigre se mogu koristiti i u svrhu ponavljanja jer izmjenjivanjem dijela gdje učenici ponavljaju sadržaj i dijela gdje se igraju povisuje njihovu motiviranost i samim time ponavljanje čini djelotvornijim. Na kraju se može zaključiti kako se videoigre mogu koristiti kao alternativa PowerPointu prilikom prezentiranja znanja, no zbog većeg vremena pripreme ga ne mogu u potpunosti zamijeniti. Međutim,

povremenim korištenjem videoigara umjesto PowerPointa se može umanjiti monotonost nastave i time povisiti motiviranost učenika.

10. Literatura

1. Allen, I. E., & Seaman, J. (2007). Online nation: Five years of growth in online learning. The Sloan Consortium: Newburyport, Massachusetts.
2. Annetta, L. A. (2008). Video games in education: Why they should be used and how they are being used. *Theory into practice*, 47(3), 229-239.
3. Apache OpenOffice (n.d.). Impress. Preuzeto 22. travnja 2017. s <https://www.openoffice.org/product/impress.html>
4. Arkorful, V., & Abaidoo, N. (2015). The role of e-learning, advantages and disadvantages of its adoption in higher education. *International Journal of Instructional Technology and Distance Learning*, 12(1), 29-42.
5. Austin, D. (2009). Beginnings of PowerPoint. Computer history. Preuzeto 22. travnja 2017. s <http://archive.computerhistory.org/resources/access/text/2012/06/102745695-01-acc.pdf>
6. Calabrese, D. (2014) *Unity 2D Game Development*. Birmingham: Packt Publishing
7. Clark, R. C., & Mayer, R. E. (2016). *E-Learning and the science of instruction: proven guidelines for consumers and designers of multimedia learning*. Hoboken, NJ: Wiley.
8. Creighton, R. H. (2013). *Unity 4.x Game Development by Example Beginner's Guide*. 3rd ed. Birmingham: Packt Publishing
9. Ćukušić, M. & Jadrić, M. (2012). *E-učenje: koncept i primjena*. Zagreb: Školska knjiga
10. De Aguilera, M., & Mendiz, A. (2003). Video games and education:(Education in the Face of a "Parallel School"). *Computers in Entertainment (CIE)*, 1(1), 1-14.
11. Enterbrain Inc (n.d.). About RPG Maker MV. Preuzeto 29. travnja 2017. s <http://www.rpgmakerweb.com/products/programs/rpg-maker-mv>
12. Epic Games Inc. (n.d.). Unreal Engine 4. Preuzeto 29. travnja 2017. s <https://www.unrealengine.com/unreal-engine-4>
13. Farkas, D. K. (2008). A heuristic for reasoning about PowerPoint deck design. In *Proceedings of the 2008 International Professional Communication Conference*. Montreal, Canada.
14. FPPT (2016). History Of PowerPoint: The Amazing Facts You Did Not Know. Preuzeto 22. travnja 2017, s <http://www.free-power-point-templates.com/articles/history-of-powerpoint-the-amazing-facts-you-did-not-know/>

15. Google (2017). About Slides. Preuzeto 22. travnja 2017. s <https://www.google.com/slides/about/>
16. Instructional Technology and Distance Learning, 12(1), 29-42.
17. Gareau, S., & Guo, R. (2009). "All work and no play" reconsidered: The use of games to promote motivation and engagement in instruction. *International Journal for the Scholarship of Teaching and Learning*, 3(1), 12.
18. Jochems, W., Koper, R., & Merriënboer, J. J. (2004). *Integrated e-learning: implications for pedagogy, technology and organization*. London: RoutledgeFalmer.
19. Lee, J. (2011). Advantages and Disadvantages of Using Powerpoint. Preuzeto 23. travnja 2017. s <https://www.apsense.com/article/advantages-and-disadvantages-of-using-powerpoint.html>
20. Liaw, S. S., Huang, H. M., & Chen, G. D. (2007). Surveying instructor and learner attitudes toward e-learning. *Computers & Education*, 49(4), 1066-1080.
21. Libre Office (n.d.). Impress. Preuzeto 22. travnja 2017., s <https://www.libreoffice.org/discover/impress/>
22. Linden Lab (n.d.). Second Life. Preuzeto 26. travnja s <http://secondlife.com/>
23. McAuley, A., Stewart, B., Siemens, G., & Cormier, D. (2010). The MOOC model for digital practice: The MOOC model for digital practice. University of Prince Edward Island.
24. Mojang (n.d.). What is Minecraft? Preuzeto 26. travnja 2017. s <https://education.minecraft.net/how-it-works/what-is-minecraft/>
25. Okita, A. (2015). *Learning C# programming with Unity 3D*. Boca Raton: CRC Press.
26. Phipps, R., & Merisotis, J. (1999). What's the difference? A review of contemporary research on the effectiveness of distance learning in higher education. Preuzeto 21. travnja 2017., s <http://www.ihep.org/research/publications/whats-difference-review-contemporary-research-effectiveness-distance-learning>
27. Piskurich, G. M. (2003). *The AMA handbook of e-learning: effective design, implementation, and technology solutions*. New York: AMACOM.
28. Prezi (n.d.). About Prezi. Preuzeto 22. travnja 2017. s <https://prezi.com/about/>
29. Richardson, J., & Swan, K. (2003). Examining social presence in online courses in relation to students' perceived learning and satisfaction. *Journal of Asynchronous Learning Networks*, 7 (1), 68-84.

30. Russell, W. (2017.). What is Microsoft PowerPoint? Preuzeto 22. travnja 2017., s <https://www.thoughtco.com/how-do-i-use-powerpoint-2767371>
31. Sherry, L. (1996). Issues in distance learning. *International journal of educational telecommunications*, 1(4), 337-365.
32. Soffar, H. (2017). The advantages and disadvantages of PowerPoint skills. Preuzeto 23. travnja 2017., s <http://www.online-sciences.com/computer/the-advantages-and-disadvantages-of-powerpoint-skills/>
33. Squire, K. (2003). Video games in education. In *International journal of intelligent simulations and gaming*. 2(1), 1-16.
34. Tait, A. (2003). Reflections on student support in open and distance learning. *International Review of Research in Open and Distance Learning*, 4(1), 1-9.
35. TalentLMS (n.d.). The history of e-learning. Preuzeto 21. travnja 2017, s <http://www.talentlms.com/elearning/history-of-elearning>
36. Tavangarian, D., Leypold, M. E., Nölting, K., Röser, M., & Voigt, D. (2004). Is e-Learning the Solution for Individual Learning?. *Electronic Journal of E-learning*, 2(2), 273-280.
37. Unity Technologies (n.d.). Unity. Preuzeto 29. travnja 2017. s <https://unity3d.com/unity>
38. Welsh, E. T., Wanberg, C. R., Brown, K. G., & Simmering, M. J. (2003). E-learning: emerging uses, empirical results and future directions. *International. Journal of Training and Development*, 7(4), 245-258.
39. Xingeng, D., & Jianxiang, L. (2012). Advantages and disadvantages of PowerPoint in lectures to science students. *International Journal of Education and Management Engineering (IJEME)*, 2(9), 61.
40. YoYo Games (n.d.). Game Maker Studio 2. Preuzeto 29. travnja 2017. s <http://www.yoyogames.com/gamemaker>
41. Zhang, D., Zhao, J. L., Zhou, L., & Nunamaker Jr, J. F. (2004). Can e-learning replace classroom learning?. *Communications of the ACM*, 47(5), 75-79.

11. Prilozi

1. Početni projekt za izradu videoigre koji sadrži i scenu primjera:

<https://github.com/pavo76/Edukativna-igra>

2. Primjer napravljene igre:

<https://drive.google.com/drive/folders/0ByOKdsc1CRcsekJLSWpWUHJpZzA?usp=sharing>